Jurassic News

Retrocomputer Magazine Anno 4 - Numero 22 - Marzo 2009



Jurassic News

Rivista aperiodica di Retro-computing

Coordinatore editoriale
Salvatore Macomer [Sm]

Redazione
redazione@jurassicnews.com

Hanno collaborato a questo numero:
Sonicher [Sn]
Tullio Nicolussi [Tn]
Lorenzo 2 [L2]
Besdelsec [Bs]
Mister X [Mx]

Impaginazione e grafica
Anna [An]

Diffusione
marketing@jurassicnews.com

La rivista viene diffusa in formato PDF via Internet agli utenti registrati sul sito www.jurassicnews.com. la registrazione è gratuita e anonima; si gradisce comunque una registrazione nominativa.

Contatti
info@jurassicnews.com

Jurassic News promuove la libera circolazione delle idee

Marzo 2009

In Copertina

Anche il Giappone partecipò all'avventura dell'home computing e lo fece prendendo le idee americane e sviluppandole secondo una logica orientale. Il Computer M5 della SORD ne è uno dei primi esempi.

Editoriale

Sempre in movimento.

Il titolo di questo editoriale non prende spunto dalla mobilità on-line, idea che prende sempre più piede e che purtroppo ha anche parecchi aspetti negativi che la maggior parte di noi finge di non vedere.
Parliamo invece un po' dei nostri progetti legati all'iniziativa del periodico dedicato all'hobby del retrocomputer in Italia.
L'anno scorso abbiamo sperimentato una periodicità delle rivista con una uscita ogni 45 giorni circa.
L'esperimento non ha avuto esito tale da consentirci di mantenere gli otto numeri/anno (questa era l'idea), per cui vi dico molto serenamente che abbiamo deciso di tornare ad un piano di uscite bimensile.

Per mantenere le uscite ravvicinate siamo stati costretti a ridurre il numero di pagine, arrivando al limite minimo di 54 mentre la media precedente era attorno alle 66/68 pagine per fascicolo e questo non ci ha soddisfatto.

Quello che il mio mandato di coordinatore mi consente di fare è lavorare per far crescere il numero di pagine della rivista (già questo numero si attesta sulle 68), cercando soprattutto di dare una continuità stabile alle rubriche.

Fra i contenuti di questo fascicolo mi piace ricordare la puntata finale delle storie della megaditta. Il nostro amico Alfonso, che ringrazio e saluto, ci lascia quale collaboratore, almeno quale "diarista" come lo abbiamo conosciuto grazie alle sue avventure. La puntata finale dei suoi racconti di vita non poteva che intitolarsi "La Fine".

[Sm]

Jurassic News

è una fanzine dedicata al retro-computing nella più ampia accezione del termine. Gli articoli trattano in generale dell'informatica a partire dai primi anni '80 e si spingono fino ...all'altro ieri.

La pubblicazione ha carattere puramente amatoriale e didattico, tutte le informazioni sono tratte da materiale originale dell'epoca o raccolte (e attentamente vagliate) da Internet.

Retrocomputing

Il ruolo di Internet nel retrocomputing

Dove si disquisisce dell'importanza della rete nell'esplicazione dell'hobby che tanto ci appassiona.

La "rete di reti", al secolo Internet, ha giocato e gioca tuttora sempre di più un ruolo fondamentale nel mondo del retrocomputing.

Dalla fondazione dei gruppi di discussione, attivi su UseNet fin dal momento della nascita dei collegamenti extra-locali, con le università americane in testa, alla creazione di siti specializzati sia da parte di semplici appassionati che di aziende "storiche" del settore, fino ai nuovi paradigmi dell'espressione (blog, podcast, etc...).

Per rendersi conto di persona di quale sia l'importanza della rete per il nostro hobby, è sufficiente fare un esercizio filosofico: la cosidetta "sottrazione di elementi". Funziona così: si ipotizza che un qualcosa sparisca nel nulla (nel nostro caso Internet) e si esaminano le conseguenze sui vari aspetti dell'attività sotto esame.

Non è difficile capire che per un retro-computerista la cosa è quanto meno fastidiosa. Significherebbe ad esempio l'impossibilità, se non fortuita, di reperire documentazione, per non parlare di software, pezzi di ricambio, sistemi completi e consigli.

Come cambierebbe la situazione, ci siamo chiesti e ne abbiamo dedotto che il nostro mondo sarebbe di gran lunga diverso.

Tanto per incominciare le collezioni sarebbero molto più ridotte. Infatti l'unico canale disponibile sarebbe quello dei mercatini e delle fiere più o meno specializzate, ma niente eBay e simiglia! E' probabile che ci si concentrerebbe su piattaforme ben definite e conosciute, deviando molto difficilmente da un sistema conosciuto. Chi era fanatico del Commodore64 magari potrebbe anche prendersi uno Spectrum, tanto per verificare personalmente le dicerie sulla superiorità dell'uno sull'altro, ma senza riuscire probabilmente ad allargare le proprie conoscenze ad altri sistemi.

La prova di questo è che non appena si viene in possesso di un computer precedentemente a noi sconosciuto, subito ci precipitiamo su Internet per cercare quanto più possibile.

Non si può contare molto sul museo, inteso come istituzione votata alla conservazione. Fanno eccezione musei "vivi" come quello per Apple realizzato dal club AllAboutApple di Savona, che mantengono e mostrano i sistemi in funzione. Un'altra lodevole iniziativa musea-

le è l'History Computer Museum in California, ma certo non è che sia proprio dietro l'angolo... Fra l'altro la nascita di questo tipo di musei, diciamo "attivi", è strettamente correlata alla disponibilità di professionisti in grado di occuparsi delle macchine.

Il canale informativo primario sarebbero le riviste, tradizionale media che da sempre ha accompagnato gli interessi delle persone.

Forse in esse vivrebbero ancora le rubriche "Vendo-Compro", per facilitare i contatti fra gli appassionati. Ma, ci chiediamo, esisterebbe un mercato sufficientemente appetibile per una rivista cartacea distribuita in edicola? Forse sì, considerando la mancanza di altre valvole di sfogo, magari anche le riviste più generali si farebbero carico di curare una rubrica dedicata ai retro argomenti.

Rimanendo nell'ambito delle riviste è evidente che la vendita degli arretrati sarebbe ancora un business interessante per gli editori. A fianco degli editori "reali" si avrebbero fanzine cartacee o su supporto, curate dai club sparsi sulla penisola. Non è che sia sparito 'sto fenomeno dei club (basta pensare ai Linux Club), ma certo sono molto diversi da un tempo.

Passiamo alla documentazione. Come sarebbe possibile reperire informazioni tecniche, manuali e software di base senza Internet? Converrete con me che, pur possibile, sarebbe di gran lunga un lavoro incommensurabilmente più faticoso. Qui ci potrebbero dare una mano le biblioteche, depositi tradizionali di materiale cartaceo. Confesso che io sono un grande appassionato di biblioteche: le considero il vero tempio della cultura e per fortuna non sono ancora contaminate del tutto dalla tecnologia e dall'efficenza a tutti i costi. Cosa che fra l'altro impedisce a queste istituzioni di buttare le vecchie riviste o libri non più consultati che sono proprio quello che andiamo cercando!

Non avremo quei (a volte) meravigliosi (altre volte un po' meno belli) siti Web curati dagli appassionati che fanno della loro raccolta retrò la loro missione.

Ma la cosa più importante è che mancherebbe quasi del tutto la possibilità per le "nuove leve" di avvicinarsi al mondo del retro computing se non attraverso una mediazione diretta. Magari il figlio di qualcuno di noi si potrebbe anche appassionare allo strano hobby del genitore, ma dicono che questo accade molto raramente.

Sicuramente non potrebbe esistere una varietà così grande di emulatori, visto che la loro realizzazione e soprattutto diffusione è possibile praticamente solo grazie alla presenza di Internet.

Aah! dimenticavo: non ci sarebe nemmeno Jurassic News :-)

[Tn]

Le prove di Jurassic News

Questa prova riguarda una macchina giapponese. Finora non ne avevamo incontrate ma il Giappone ha pieno diritto di occupare un posto da protagonista nella nascita ed evoluzione dell'informatica personale.

SORD Computer - M5



Una immagine "plain" della macchina. Le dimensioni reali sono leggermente superiori a quelle dello Spectrum.

Contesto storico

Il sistema M5 della Sord Computer Corporation esce nel 1982 e si pone nella fascia home computer a basso prezzo; la stessa delle macchine Sinclair e Commodore per capirci.

Non è il primo computer costruito dalla Sord, che ha iniziato la propria avventura nell'elettronica digitale producendo macchine da ufficio, ma è il primo tentativo di copiare per il mercato giapponese e perché no mondiale, il successo delle "macchinette" a bassissimo costo e ad altrettanto basso potenziale elaborativo.

All'esterno del Giappone il Sord M5 non ha venduto un granché, se si eccettua un mercato cecoslovacco abbastanza florido, sempre considerando l'economia di un paese dell'Europa dell'Est. Forse per motivi contingenti (ad esempio un importatore intraprendente o accordi politici fra i due paesi) o forse per un test di accettabilità del sistema, quello cecoslovacco rimase comunque solo un tentativo di penetrazione nel mercato europeo già occupato dai colossi Sinclair e Commodore.

Si attribuisce comunemente il successo cecoslovacco del Sord M5 per il fatto che la catena di negozi che lo distribuiva, la TUZEX, aveva inventato una specie di cambio della moneta che permetteva di acquistare la macchina con "dollari virtuali" con un rapporto 1 a

5 rispetto alla moneta ufficiale (la corona cecoslovacca).

In Inghilterra viene ri-pachettizzato dalla Computer Games Limited (CGL) come CGL M5, ma la sua diffusione è stata scarsa nonostante il prezzo in linea con lo Spectrum e le prestazioni paragonabili con il vantaggio di una qualità costruttiva decisamente superiore.

La Sord venne fondata nel 1970 dall'allora ventisettenne Takayoshi Shiina. L'informatica era una tradizione familiare dal momento che la madre era una programmatrice della piattaforma PDP della Digital e lo stesso fondatore della Sord lavorava come commerciale per una succursale Digital in Giappone.

La prima idea della ditta era quella di entrare nel mercato business con macchine da ufficio (M23, M35), ma la disponibilità di progetti di basso costo unita ad una domanda crescente di macchine home, la convinse ad accettare la sfida dei vari Spectrum proponendo un sistema abbastanza simile al rivale Sinclair.

La Sord fece seguire al primo progetto ulteriori issues chiamate M5 Pro e M5 Jr, inglobando l'alimentatore e potenziando la dotazione iniziale della macchina. Dal punto di vista del software la Sord si era ritagliata un mercato business inventando il PIPS, una sorta di foglio elettronico programmabile con un linguaggio macro. Una versione adeguata all'M5 venne pachettizzata su cartridge, così come numerosi titoli ludici che andavano per la maggiore all'epoca.

L'espansione del sistema viene garantita da una serie di prodotti distribuiti dalla stessa Sord che vanno da uno slot multiplo per le cartridge all'unità floppy da 3.5", immaginiamo a costo proibitivo.

L'unità base venne venduta in Inghilterra ad un prezzo attorno alle 150 sterline, più o meno il prezzo dello Spectrum non espanso. Come spesso succede però poi bisogna comprare qualche espansione per lavorare meglio... e il prezzo inevitabilmente sale.

SORD m5
CREATIVE COMPUTER
Why your family will want one . . . or two!
M5 STANDARD SPECIFICATIONS
ORDER NOW FOR CHRISTMAS

La pagina pubblicitaria su Bits & Bytes, una rivista della Nuova Zelanda.

L'indispensabile per lavorare: unità centrale, alimentatore e interprete BASIC con manuale (manca solo il monitor e siamo a posto).



Primo approccio

Del Sord M5 ne esistino almeno tre versioni che si diffenziano per la colorazione parziale del case (che va dal bianco sporco al giallo ocra) e per il il layout della tastiera, localizzata per il mercato di riferimento.

Il sistema arriva in una scatola di cartone che comprende unità centrale, alimentatore e manuale. La confezione (che abbiamo riprodotto in trasparenza su questa pagina) è spartana ed elegante al tempo stesso. Sul coperchio il nome del produttore e il logo del prodotto il cui vero nome sarebbe "m.5", ma ovunque è conosciuto come "M5" ed è la sigla che useremo anche noi.

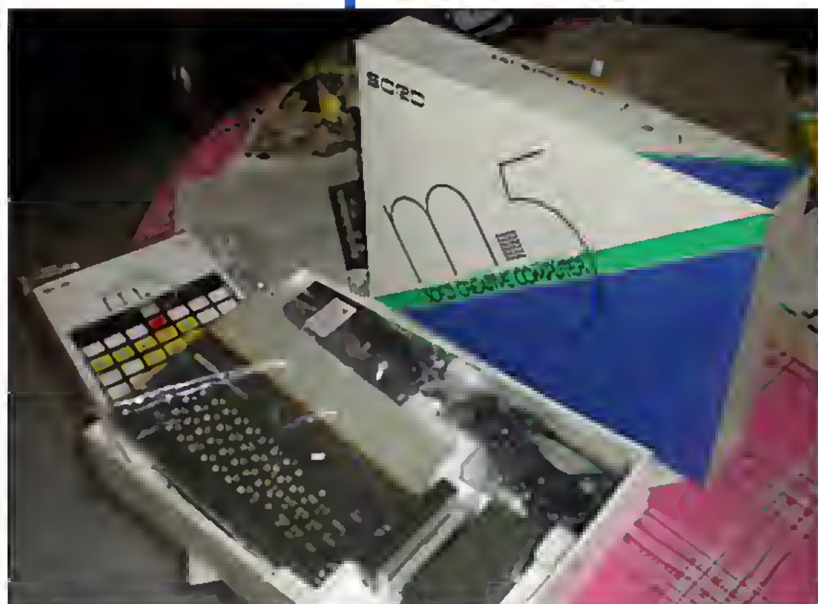
In generale il sistema somiglia in modo sorprendente allo Spectrum, anche se è leggermente più grande e soprattutto di spessore maggiore del rivale Sinclair e presenta due fasce distinte per colore. Le dimensioni sono 262 x 185 x 36 mm. , il peso di 800 grammi al quale si somma l'alimentatore esterno, una generosa scatolotta nera, dal peso di 1 Kg circa.

La parte tastiera è sempre di colore nero o grigio molto scuro con tasti dello stesso colore stampigliati in bianco e arancio (per le funzioni alternative). Contiamo 55 tasti, alcuni dei quali sono "speciali", come il Reset, due Shift, il Control, etc... La parte superiore è quella colorata diversamente con contrasto chiaro che riporta la sigla del sistema (un M5 stilizzato) sul lato sinistro, lo stesso colore è riservato al guscio inferiore, base del sistema. In realtà la banda chiara è un coperchio che nasconde lo slot di espansione per le ROM, per cui nell'uso normale sarà sempre in posizione verticale costituendo una specie di leggio stampigliato con le funzioni più comuni.

La tastiera è ovviamente la parte che salta subito all'occhio dell'esaminatore. Assomiglia molto a quella dello Spectrum, anche se i tasti sono quadrati con un vezzoso taglio dell'angolo in basso a destra, a parte i quattro all'estrema destra che hanno il taglio in basso a sinistra, e consentono un feedback migliore sotto le dita. I tasti sono di gomma, come quelli dello Spectrum, ma con una corsa inferiore e "meno dura" che consente una digitazione accellerata seppure con meno certezza nel typing. Gli utilizzatori sono concordi nell'affermare che la tastiera ha una qualità migliore ed è anche notevolmente più robusta rispetto a quella dello Spectrum.

Anche nella disposizione dei tasti

Il contenuto della "scatola magica"



si vede un approccio comune con il rivale di riferimento, ad esempio per lo space, relegato sulla destra come un tasto normale, seppure di dimensioni superiori. L'approccio software seguito dalla Sord non nasconde il plagio con lo Spectrum ad esempio nell'utilizzo di un Basic monofunzione, cioè con gli statement attivabili da un unico tasto e digitazione guidata dalla tipologia del cursore (L, K, etc...).

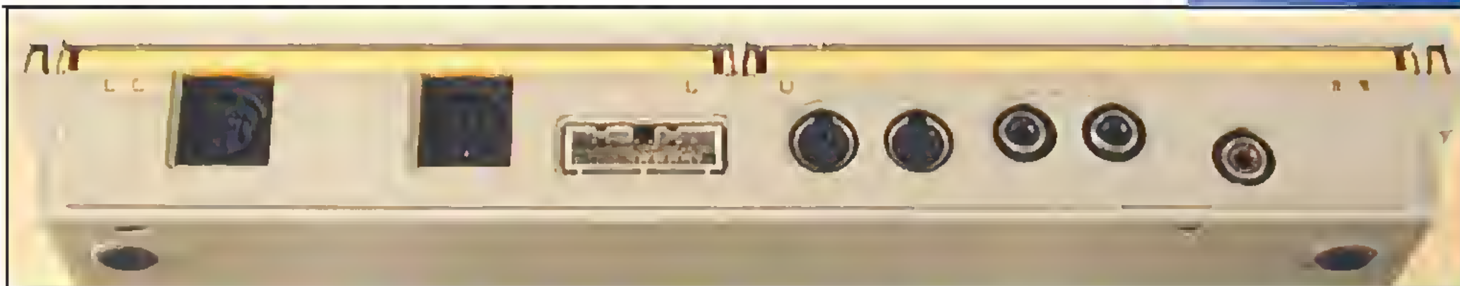
Un led rettangolare rosso posto sulla parte alta a destra della tastiera, ci avvisa dell'alimentazione del sistema.

Continuando nell'esame esterno scopriamo sul retro la fila di connettori che permettono al sistema di colloquiare con l'esterno. Da sinistra troviamo il connettore DIN a sei poli dell'alimentatore; il connettore DIN a 8 poli del registratore a cassette; una parallela Centronics con attacco piatto 8+8 pin; Due connettori mini-din sono usati per i joystick o le paddle; seguono due connettori jack per il monitor e l'out audio ed infine l'attacco tipo "antenna" per il TV domestico.



Più convincente della precedente la pagina del depliant fatto circolare in Giappone. Si noti che il nome ufficiale del computer è "m.5".
Sotto: l'm5 con cartridge inserita.





Il retro della macchina con le uscite e i connettori per le periferiche. Da sinistra: alimentatore, registratore a cassette, parallela, 2 joystick, audio e video, modulatore TV.

Hardware

Il sistema M5 è progettato attorno ad una CPU Z80A a 3.58 MHz con un chip custom, un CTC (Timer-Counter) e un generatore video della Texas Instruments TMS9929.

Nella versione originale trova spazio una EPROM da 8 Kb e una RAM da 4 Kb che si unisce ai 16 Kb utilizzati per il video. Le modalità video sono abbastanza buone con due modi definiti: testo, che da 24 righe da 40 caratteri in una matrice 8x6; grafica che rende disponibili 256x192 pixel a 16 colori ed eventualmente 32 sprite da 16x16 pixel, controllati in hardware.

Le pagine testo sono addirittura due e si possono usare e switchare a proprio piacere.

L'uscita video prevede il classico TV Out modulato per il televisore e una uscita monitor che bypassa il modulatore.

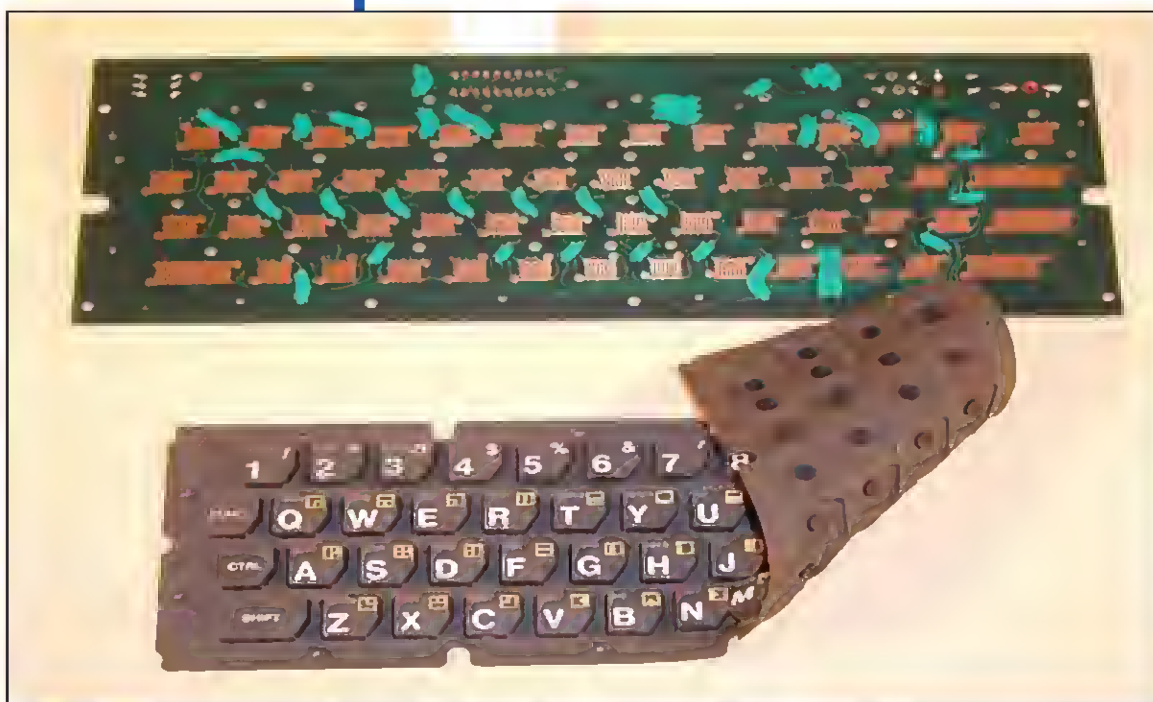
Il suono è anch'esso in linea con i migliori progetti home del momento: 3 canali da 6 ottave più uno di rumore, controllati dal generatore di segnali SN76489AN, sempre di fabbricazione Texas Instruments

Le porte di I/O sono completate da una centronics parallela, dall'uscita per registratore (DIN a 8 pin) e da due connettori per Joystick/Paddle di fabbricazione proprietaria.

Il registratore a cassette (ne esiste una versione ufficiale della Sord), si interfaccia con il sistema alla velocità di 2 Kbit/sec (nella pratica si tratta di una interfaccia a 1200 boud).

Notevole la presenza dello slot di espansione per ROM, accessibile dallo sportellino superiore e costruito in modo da essere decisamente più robusto rispetto

La tastiera smontata. Come si vede è decisamente di gomma...



alla soluzione "User Port" dei sistemi Sinclair/Commodore. D'altra parte si perde una certa propensione verso l'autocostruzione di interfacce e periferiche da accoppiare attraverso il classico e poco costoso pettine degli altri home.

Le possibilità di espansione della RAM permettono di arrivare a 36 Kbyte e a 16 Kbyte per la ROM.

L'unico connettore di espansione si "estende" all'utilizzo di più di una cartridge mediante duplicatori da comperare a parte.

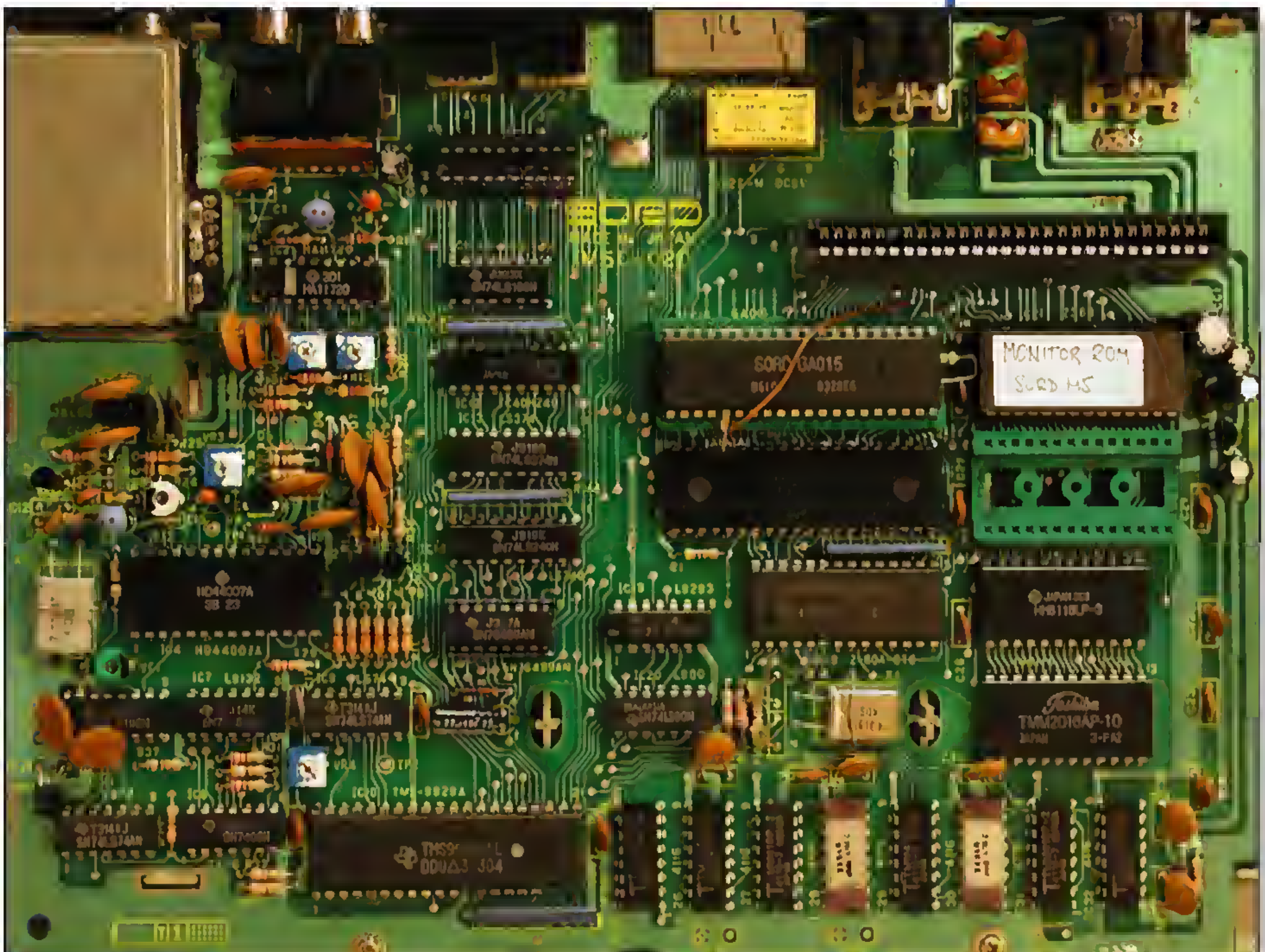
L'alimentatore esterno si collega tramite un connettore DIN a sei poli e genera le tensioni (+5, -12, +12) necessarie all'elettronica.

Qualcuno ha ipotizzato che l'M5 sia stato un precursore dello standard MSX e infatti ha caratteristiche che si ritrovano nello standard che uscirà qualche anno più tardi (Slot per espansione, CPU, suono, etc...).

La Sord ha dotato il sistema M5 di periferiche dedicate che ne espandono le funzionalità. Fra queste: Cartridge multiplexer (EC-5), 32 KB RAM expansion (EM-5), joysticks (JS-5), joypads (JP-5), expansion box (EB-5), thermal printer (PT-5), Parallel I/O cartridge (PI-5), Serial interface cartridge (SI-5), Floppy disk drive (FD-5).

Il multiplexer (prodotto EC-5) è praticamente indispensabile per

La piastra madre, ordinata e dall'aspetto molto professionale, a parte una correzione "al volo" su uno dei pin del chip custom.





L'unità floppy disk da 3.5 pollici.

usare uno dei BASIC più avanzati (BASIC-G e BASIC-F) che richiedono memoria aggiuntiva. In questo caso le due cartridge dell'interprete scelto e dell'espansione di RAM da 32 Kb trovano posto nei due slot disponibili all'espansione.

In caso siano richieste ulteriori espansioni, come ad esempio se si possiede l'unità floppy che ha bisogno di una interfaccia, è disponibile un box di espansione da agganciare sul retro dell'M5. Viene chiamato Expansion Box e consente appunto di avere più ROM collegate; il sistema provvede a switchare gli indirizzi nel caso di utilizzo dell'una o dell'altra periferica.

La cartridge seriale mette a disposizione una interfaccia RS232C programmabile mentre è disponibile una ulteriore parallela, oltre la Centronics built-in, completamente programmabile in I/O.

La massima espansione si ottiene acquistando l'unità floppy (Intelligent Floppy System). Si tratta di un supporto da 3.5" che permette di memorizzare fino a 190 Kbyte di informazioni. Sembra inve-

Alcune periferiche del Sord M5











ce che manchi del tutto un sistema operativo adeguato, anche se i BASIC avanzati permettono di scrivere e di leggere file dall'unità.

Accanto ai classici joystick a cloche, la SORD vende due Joypad, che sono costituiti da una rotellina cliccabile nelle otto dimensioni più il classico punsante Fire.



Un M5 al lavoro, con Basic-I e registratore dedicato.

Uso

La cosa più scomoda del Sord M5 è che viene venduto senza il BASIC in ROM e di conseguenza è sempre necessario operare con una ROM aggiuntiva (e cassetto aperto).

Quello che è contenuto nella ROM di sistema da 8 Kb è il cosidetto Monitor e ovviamente la mappa dei caratteri.

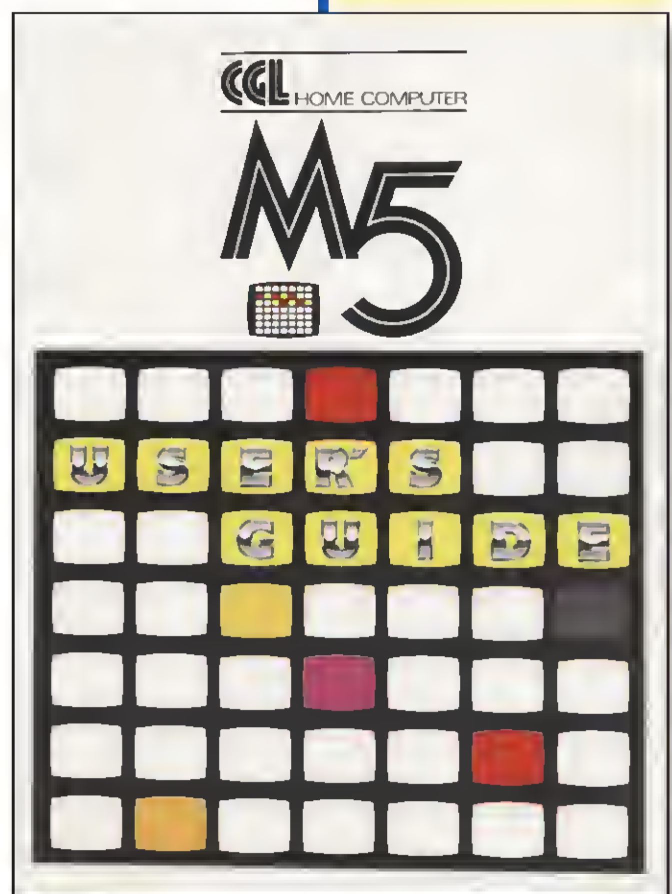
All'avvio del sistema, se presente una ROM, la macchina fa il boot con il codice contenuto nella cartridge: se si tratta di un gioco questo viene lanciato, se si tratta di uno dei BASIC disponibili ci si ritrova nell'ambiente dell'interprete. Se invece la macchina è "nuda", cioè senza espansioni, ci si ritrova sotto il "monitor" di sistema al quale fra l'altro è dedicato un manuale che arriva assieme alla macchina.

Con questo monitor si può fare pochino: giusto inserire qualche istruzione in esadecimale, esaminare la memoria e lanciare il codice da una certa locazione.

La macchina viene fornita con due manuali base: lo User's Guide e il già citato Monitor Handling Manual. Acquistando i pacchetti aggiuntivi, ad esempio un interprete BASIC, si avranno i corrispondenti manuali.

Prendendo confidenza con i comandi principali scopriamo che il reset si ottiene con la combinazione

La User's Guide, qui in versione della CGL che l'ha venduto in Inghilterra come macchina da gioco.





ROM e manuale della versione "scientifica" dell'interprete.

dei tasti CONTROL e RESET, appunto. SI tratta di un break hardware che interrompe qualsiasi esecuzione in corso e rimanda all'ambiente di partenza (Monitor o BASIC).

La digitazione è piacevole con un feedback tattile e la bassa corsa dei tasti che ne favoriscono la veloce digitazione. E' attivo l'auto-repeat, funzione comodissima ma spesso trascurata proprio sulle macchine prive di tastiera meccanica che ne avrebbero invece un grande bisogno!

Le impressioni di utilizzo restituiscono un sistema "onesto" nelle prestazioni ma senza particolari picchi. Per averne una impressione migliore e scoprire quanto si discosta dai più famosi "fratellini", cioè dai vari Spectrum e Commodore, bisogna sporcarsi le mani con i due BASIC avanzati e quindi con grafica e suono.

Software

Il Sord M5 viene venduto con il BASIC-I su ROM cartridge. Opzionalmente possono essere acquistati altri due interpreti: BASIC-G e BASIC-F. Le lettere che seguono il nome dell'interprete hanno i seguenti significati:

I = Initial, definito "Easy BASIC for Beginners"

G=Graphics/Game, "Easy BASIC for Games"

F=Scientific, "Floating Point"

Ognuna di queste cartridge costa all'incirca 35 sterline, cioè circa 120-150 mila Lire se fossero state importato in Italia.

L'output, qui ottenuto con un emulatore per esigenze editoriali.

```
Ready
list
10 PRINT "Emulator Sord m5"
20 PRINT "Ono to skutecne slape!
"

Ready
run
Emulator Sord m5
Ono to skutecne slape!

Ready
```

L'interprete della SORD è allineato con le prestazioni della concorrenza del periodo. Le parole chiave si possono inserire con tasti chiave, come nello Spectrum, o digitandole per esteso. Per l'editor delle linee troviamo

l'Auto e il Renumber. Per la lettura del device Joystick/Joypad è disponibile l'istruzione dedicata Joy.

La SORD Computer è anche famosa per aver inventato il PIPS, una sorta di super spreadsheet programmabile che, a loro dire, "permette di utilizzare il computer senza saper programmare" (da una pubblicità apparsa su Elettronica Oggi nel dicembre 1982). Per la serie M5 in realtà il PIPS viene semplificato e venduto con il nome di FALC.

La SORD ha messo molta enfasi sulla disponibilita del package FALC che è una sorta di spreadsheet le cui celle sono informazioni più generaliste rispetto ai classici VISICALC e similia che sono il riferimento. In realtà senza l'espansione di RAM a 36Kb, che rende la disponibilità di "celle" al programma dell'ordine di 8x60, si può fare pochino.

Il BASIC-I è molto "basilare", appunto. Si tratta di un interprete senza capacità grafiche, se non quelle possibili combinando i caratteri semigrafici disponibili sulla tastiera.

Il BASIC-I riconosce solo l'aritmetica intera (una scelta che troveremo oggi inaccettabile ma che non sembrava scandalizzare troppo gli utilizzatori dei primi anni '80), Il BASIC-G contiene funzioni grafiche per la tracciatura di linee, cerchi e rettangoli nei colori scelti, compreso il riempimento di un'area chiusa



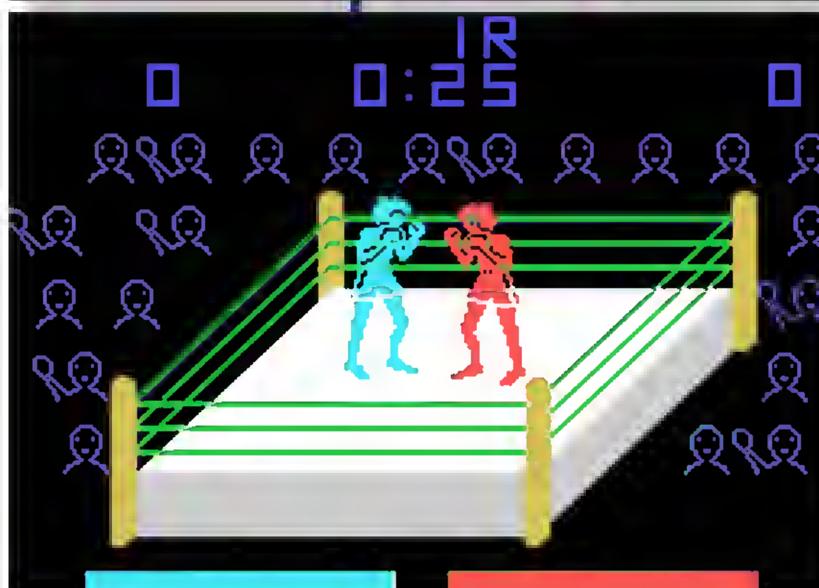
con il comando PAINT.

La gestione degli sprite consente la realizzazioni di giochi con funzioni facilitate per quanto riguarda il movimento e la collisione delle figure a video. Uno sprite può essere definito come un'area di 16x16 pixel o di 8x8 pixel e possono anche essere combinati assieme per formare figure più articolate. Gli sprite possono essere mossi, nascosti, invertiti, etc... con semplici istruzioni disponibili nel BASIC-G. Oltre agli sprite è definibile l'area di sfondo, tracciabile a piacere. Dato che gli sprite possono stare su 32 livelli diversi (nel qual caso non collidono), è facile realizzare movimenti di figure contro uno sfondo fisso o scrollabile.

Le capacità grafiche della macchina sono articolate ma non proprio di facile gestione. L'M5 ha quattro modi video, se andiamo ad espandere le due modalità di base: text, GI, GII e un modo chiamato multicolor. Queste modalità si disputano l'organizzazione dei 16 Kb di RAM







Il Sord M5 si dimostra all'altezza dei concorrenti come macchina da gioco.

video con le conseguenti limitazioni di definizione e utilizzo dei colori. Ad esempio nel modo GII si hanno a disposizione 192x256 pixel ma solo con due colori per ogni unità definita come un'area di memoria da 8 pixel, mentre se si vogliono usare tutti e 16 i colori disponibili bisogna stare su una risoluzione di 48x64 pixel.

Sul fronte del suono è lo statement PLAY che consente la parametrizzazione del generatore sonoro a tre canali. La dotazione di parametri è notevole con la presenza di effetti sonori, l'impostazione del volume, etc...

Per quanto riguarda la programmazione si possono definire nomi di variabili fino a 32 caratteri, tutti significativi; notiamo la presenza di Label (nomi che possono essere usati come marcatori di riga e richiamati dalle istruzioni di GOTO e GOSUB). Notevolmente si trova una gestione degli interrupt abbastanza inconsueta per questa classe di macchine. In pratica al verificarsi di un dato evento è possibile impostare la continuazione del programma da una data label:

```
100 ON ALARM GOTO 200
300 ON KEY GOSUB 320
...
```

Gli eventi tracciabili sono legati al timer (ALARM), alla pressione di un tasto (KEY) o di un segnale dai joypad (PAD) e al verificarsi di una collisione fra sprite (COINC). Infine il parametro ERROR interrompe l'esecuzione al verificarsi di una condizione di errore nell'interprete.

Conclusioni

E' un peccato che il SORD M5 non abbia trovato diffusione nell'Europa occidentale, anche se le sue caratteristiche poco si discostavano da quelle degli home che all'epoca andavano per la maggiore. Un peccato perché un sano confronto concorrenziale avrebbe potuto, arricchendo l'offerta, slegarci un pochino dal dualismo Commodore vs Sinclair (sempre lasciando Apple fuori dalla classe degli home a basso costo).

Se andiamo ad esaminare le caratteristiche tecniche del sistema e della sua dotazione hardware/software, diciamo che di positivo troviamo:

- *un look più accattivante ed "elegante" rispetto alla concorrenza;*
- *una estesa possibilità di programmazione nel comparto giochi per le caratteristiche plug-in (sprite, etc...) e per la gestione delle periferiche di input (joystick, joypad) già compresa nella dotazione base e ben supportate dall'interprete BASIC-G;*
- *un BASIC qualche volta più veloce ed ottimizzato con istruzioni di sviluppo (Auto, Renumber) che spesso latitavano nei corrispondenti prodotti della concorrenza;*
- *la presenza di interfaccia parallela Centronics e di output video+monitor, anche queste diventate poi standard di fatto.*

Fra gli aspetti positivi possiamo citare anche il prezzo (attorno alle 100 sterline) che è in linea con la concorrenza e la capacità di espansione con periferiche dedicate (ovviamente queste fanno lievitare notevolmente il costo del sistema).

Gli aspetti negativi che abbiamo rilevato sono invece:

- *la scarsa dotazione di RAM iniziale (appena 4 Kb) con conseguente necessità di procurarsi subito l'espansione a 32 Kb;*
- *la mancanza dell'interprete, anche quello base, in ROM;*
- *un linguaggio BASIC diverso rispetto ai più classici Microsoft, cosa che rende più difficoltosa la migrazione dei sorgenti da una piattaforma all'altra;*
- *la sostanziale trasparenza della macchina sulle riviste del settore a causa della sua scarsa diffusione; questo significa che l'appassionato alle prime armi trova meno codice da copiare e meno spunti sui quali affinare le proprie conoscenze.*

Difficile quindi tirare una conclusione assoluta. Da parte nostra il giudizio è comunque positivo perché siamo attratti dall'accattivante design e dalla qualità costruttiva dell'insieme. Per il resto è un home in linea con le prestazioni di uno Spectrum o di un Commodore 64.

[Tn]

Come eravamo...

La storia dei sistemi e degli uomini che hanno creato un mondo nuovo.

Takayoshi Shiina



Il fondatore della SORD Computer Corporation, il giapponese Takayoshi Shiina.

Come complemento dell'articolo-prova del computer SORD M5, riportiamo il testo di una intervista del giornalista Terrie Lloyd, della testata Computer Japan, al fondatore della società SORD, Takayoshi Shiina, apparsa sul fascicolo dell'agosto 1994. Il testo originale è in inglese (per fortuna non in giapponese!) e abbiamo pensato di farne una traduzione per rendere meglio l'atmosfera di qualche anno fa e scoprire i risvolti storici della nascita e declino della SORD Computer fondata da Shiina.

L'intervista è stata fatta nel 1994. Si parla quindi di una decina d'anni dopo le vicende SORD, un tempo abbastanza lungo per aver stemperato le emozioni ed avere permesso una analisi di quegli anni al fondatore della ditta. Contemporaneamente si parla del mercato computer in Giappone, mercato del 1994, delle sue prospettive future e delle sfide ed opportunità che si presentano dall'esplosione del fenomeno dei cloni IBM.

Cj: parliamo della tua prima esperienza di business con la società SORD. Quando hai fondato la SORD Computing e a che età?

Ts: fondai la SORD negli anni '70 ed avevo 26 anni.

Cj: SORD è stata una azienda di successo nei primi anni '80, vero?

Ts: Sì, possiamo ben dirlo. In particolare il 1983 è stato il nostro anno migliore con un fatturato di 22 miliardi di yen che salgono a 35 miliardi comprendendo le succursali dell'azienda in Irlanda, Singapore, Cina e Nuova Zelanda. I dipendenti totali erano circa 1800.

Cj: La crescita della SORD è stata rapidissima, ma anche la crisi visto che già nel 1984 incontro gravi problemi. Cosa successe?

Ts: Quasi tutti i miei problemi derivarono dalla mia lingua lunga! Feci una serie di annunci che poi dovetti rimangiarmi... A parte questo credo di essere stato oggetto delle mire di una imprenditorialità spregiudicata. Infatti quando l'azienda andava

benissimo, nel febbraio 1983, ricevetti una telefonata dal presidente di una grande azienda giapponese che mi propose di vendergli la mia azienda. Rifiutai, ma successivamente le sue insistenze si fecero sempre meno velate. Mi diceva: -"Suvvia Shiina, vendi adesso che lo puoi fare!". Io non capivo perché avrei dovuto chiudere un business che rendeva molto ma soprattuutto che stava crescendo ad un tasso a due cifre. In pratica sottovalutai i "consigli" di questa persona. Qualche mese più tardi cominciarono a succedere delle cose strane come i ritardi nella consegna da parte dei fornitori, componenti indispensabili per la costruzione dei nostri calcolatori che improvvisamente sparivano dal mercato o si facevano improvvisamente più costosi, le banche che si rifiutavano di estendere il credito,... insomma cose apparentemente slegate ma che con il senno di poi capisco che erano parte di un preciso disegno.

Cj: Così tu pensi che qualcuno esterno all'azienda aveva interesse a farla fallire?

Ts: ne sono assolutamente sicuro: era una campagna orchestrata molto bene e condotta da gente molto preparata e priva di scrupoli. Essi contattavano i miei clienti più grandi dicendo cose come: -"La SORD ha problemi finanziari", così finimmo in bancarotta nel giro di pochi mesi. Questi "rumors" cominciarono a venire diffusi sui periodici e giornali più seguiti come il quotidiano "Nikkey newspaper". Per una azienda in crescita rapida come la nostra era normale utilizzare il credito bancario e quindi essere esposta molto su questo fronte e dipendere in maniera sostanziale dal credito. Quando le banche cominciarono a non mantenere gli impegni presi con noi in precedenza, non avemmo modo di arginare la crisi.

Cj: ma c'era della verità in queste storie?

Tc: Assolutamente no! Il nostro profitto nel solo 1983 fu di ben due miliardi di yen, potevamo mantenere senza difficoltà tutti gli impegni finanziari sottoscritti. Dopo le indiscrezioni sulle nostre presunte difficoltà finanziarie cambiò tutto; è stata una esperienza terribile. Il nostro paese quando decide di "tagliare le corna alla lumaca" lo fa in maniera definitiva e molto "giapponese".

Cj: Hai imparato qualcosa da questa esperienza?

Ts: prima di tutto che devi avere una parte dei mass media dalla tua parte con i quali zittire ogni tentativo di insinuazione o altre manipolazioni che riguardano la tua azienda. Nell'azienda dove opero attualmente controlliamo alcune pubblicazioni proprio per avere questo canale di controllo.

Secondo: è indispensabile possedere una quota maggioritaria consistente di azioni dell'azienda,

proprio per prevenire attacchi speculativi che consentano a terzi di tagliarti i canali di finanziamento.

Cj: Ad un certo punto la SORD fu venduta a Toshiba. Ci guadagnasti nella vendita?

Ts: Assolutamente sì: con i miei partner dividemmo i 3,5 miliardi di yen della vendita. Purtroppo la SORD era una azienda famigliare e la legislazione corrente imponeva che sulla vendita di questo tipo di società fosse applicata la tassa maggiore. Pagai l'85% di tasse! Ora la legge è cambiata ma per me è troppo tardi. Ho una sola cosa da rimproverarmi: non avevo un "piano B" in testa, ma all'epoca si può dire che ero abbastanza ingenuo. I miei unici pensieri erano di lavorare sodo e far crescere l'azienda. (ride).

Cj: C'era un legame fra la Toshiba e la gente che ti aveva screditato nel 1984?

Ts: Questa è stata una insinuazione fatta dai giornali, ma da parte mia non ho mai dichiarato questo e anzi sono sicuro che Toshiba non c'entrasse per nulla nelle mie sfortunate vicende.

Cj: Lasciasti la SORD subito dopo averla venduta?

Ts: Oh no. Lavorai ad un piano di ricostruzione della società che avrebbe consentito di riportarla in attivo nel giro di due o tre anni. Sapevo quali erano stati i problemi e li esposi alla direzione della Toshiba. Sfortunatamente questi erano vecchi manager legati ad una idea tradizionalista di business e non accettarono le mie proposte. Questo mi rese piuttosto nervoso e impaziente di intraprendere una mia nuova iniziativa, così lasciai dopo circa due anni.

Cj: quando fondasti la Proside?

Ts: nel giugno 1987 e nell'ottobre dello stesso anno cominciò ufficialmente il suo business.

Cj: cominciando per la seconda volta, lo feci con dei partner?

Ts: Sì, due dei miei collaboratori più stretti mi seguirono dalla SORD: Mr. Nobuhiro Sate e Mr. Akihito Hiroishi. Senza partner me ne sarei andato negli Stati Uniti per tentare da li la fodazione di una nuova azienda, ma questi due fantastici amici mi convinseo a rimanere in Giappone.

Assieme avevamo fondato la SORD. Sate è stato il creatore del primo sistema operativo per PC in Giappone; Hiroishi proveniva dalla Sharp ed era stato il primo a progettare un PC commerciale in Giappone: era un sistema basato su 8080, una macchina del 1972!

Cj: Quale era la situazione del mercato nel 1987 quando fondaste la Proside? Mi ricordo che tutti usavamo il DOS 2.11.

Ts: Non c'erano cloni del PC IBM in giro e la NEC era la casa di riferimento per il mercato dell'elaborazione business. Nessuno pensava di sfidarla. Io trovai che questa situazione poteva essere una grande opportunità. La mia strategia con la Proside era molto diversa da quella che avevo ideato per la SORD. Noi eravamo una piccola compagnia e invece che pensare di fare tutto da soli pensai di convincere investitori stranieri ad entrare nel mercato dei PC in Giappone fornendoci il materiale per assemblare un PC compatibile. Pensai di essere stato molto fortunato a scoprire una azienda che si chiamava Tomcat che aveva sviluppato un PC AT che poteva eseguire anche il codice scritto per il NEC della serie 9800. Non era facile convincere i clienti di NEC che il nostro sistema dava migliori prestazioni, e fummo aiutati in questo da una massiccia campagna pubblicitaria. Alla fine ci mise lo zampino la sfortuna: la Tomcat non riuscì a mantenere gli impegni e il progetto dovette essere abbandonato per concentrarci solo sul mercato IBM.

Cj: Quanto valeva il mercato?

Ts: Fra IBM originali e cloni, nel 1987 si arrivava attorno a 10.000 macchine all'anno. La Proside ne importò e vendette circa 800: un successo per il primo anno, visto che eravamo operativi praticamente solo da ottobre. I nostri clienti migliori erano le aziende giapponesi che producevano stampanti e altre periferiche: ce ne compravano a decine di PC cloni e la Proside non chiuse mai un bilancio in rosso.

Cj: e dopo il 1987 cosa accadde?

Ts: Era un periodo ideale per la crescita di piccole compagnie come la nostra. Nel 1988 prendemmo una macchina chiamata AX che avrebbe dovuto competere con il mercato NEC in maniera diretta. Io non creditti mai a ciò per il semplice fatto che tutte le componenti erano proprietarie: tastiera, video, etc... Avevo ragione.

Poi nel 1990 IBM propose il suo sistema operativo DOS/V, un multiutente che poteva competere con il sistema NEC. Ero molto eccitato della cosa e lavorai per inserire la Proside in questo tipo di offerta.

Cj: E oggi, come è organizzata la tua azienda?

Ts: Ho 45 collaboratori fissi e una fabbrica con una trentina di addetti. Il nostro fatturato va dai 2 ai 2,4 milioni di dollari al mese e vendiamo circa 8000 PC all'anno. Io lo valuto attorno al 2% dell'intero mercato DOS/V in Giappone.

Cj: Come vedi il mercato in Giappone nei prossimi due anni?

Ts: Credo che il mercato dei PC e specialmente del settore multiutenza dove si inserisce il DOS/V, abbia un grande futuro. Mi aspetto una crescita del 40% quest'anno (1994 ndr) e l'anno prossimo do-

vrebbe attestarsi attorno alle 800-900 mila macchine, un terzo di tutti i PC venduto in Giappone.

Cj: pensi che NEC perderà quote di mercato?

Ts: Sì, la NEC venderà più o meno lo stesso numero di sistemi, ma il mercato crescerà e la loro percentuale inevitabilmente sarà ridotta. Quest'anno il loro share di mercato dovrebbe essere attorno al 47% e il prossimo anno scenderà al 43% e al 40% fra tre anni. Non credo scenderà sotto questa soglia perché NEC metterà in campo nuove strategie per difendersi. Nessuno può predirre cosa succederà dopo.

Cj: e di Apple cosa ne pensi?

Ts: La Apple è alla sua quota massima di mercato: il 15% circa. Il loro mercato è quello degli utenti di classe media ma ora questi hanno già comprati il Machintosh e la quota di Apple nel mercato PC è destinata a stabilizzarsi, probabilmente verso il basso. Il mercato si sta espandendo ma non nella direzione dove Apple è forte. Si venderanno sempre nuovi PC alle classi meno abbienti, mentre chi può spendere per comprare un Apple non contribuirà significativamente alla crescita del mercato.

Cj: Che ci dice in merito alle dimensioni e all'espansione del mercato in Giappone?

Ts: nell'ultimo anno il numero di macchine vendute è stato circa di 2,1 milioni di unità. Il prossimo anno mi aspetto che sfiori i 3 milioni di unità e forse i 4 milioni l'anno successivo.

Il fatto è che quest'anno circa 2,5 milioni di persone in Giappone hanno comprato un PC e ancora di più sono stati coloro che hanno acquistato un wordprocessor. Molti wordprocessor sono stati presi da gente che non ha un PC adeguato e che probabilmente sta aspettando che calino i prezzi ed aumentino le prestazioni per acquistare una nuova macchina. Si comincia ad intravvedere un mercato del secondo PC, mentre finora la gente ha acquistato il primo PC della sua vita.

Cj: Tornando a parlare di aziende, tu adotti un qualche schema incentivante per aumentare le performance dei tuoi impiegati?

Ts: No, nessun programma. Ci abbiamo provato qualche trimestre fa ma il risultato è stato un calo di efficienza invece che un aumento! Il lavoratore giapponese preferisce sentirsi parte di un gruppo piuttosto che correre da solo. Penso che questo "luogo comune" sul lavoratore giapponese sia molto vero. Pensavo di tentare la strada delle stock option ma le esperienze dimostrano che l'impiegato che riceve delle stock option si precipita a venderle sul mercato e questo non contribuisce alla stabilità

dell'azienda che ne vede il valore scendere. Noi preferiamo pagare direttamente gli impiegati con un dividendo sugli utili. Attualmente distribuiamo un 10-20% di utili in incentivi e i collaboratori sono più contenti di ricevere meno valore nominale rispetto a delle azioni ma avere i soldi in banca subito disponibili.

Cj: State cercando degli investitori o dei partner commerciali?

Ts: Sono aperto alle collaborazioni e soprattutto alle join venture. Pensiamo di poter offrire un valore aggiunto alle compagnie che intendessero entrare nel mercato giapponese con il nostro database di clienti consolidati che si aggira sui 100.000 contatti. Quanche volta cerco di convincere i banchieri che questa è una ricchezza "invisibile" dell'azienda, ma loro non mi credono (ride).

Cj: Dove vedi la tua azienda fra due o tre anni?

Ts: penso che continueremo ad occuparci dello stesso business, ampliando i nostri store di 20-30 unità. Stiamo concentrandoci sul mercato degli ordini postali dove pensiamo di avere buone chance per il futuro.

Cj: Ci sono nuovi imprenditori che si affacciano al mercato del computer in Giappone?

Ts: Credo che l'azienda Softbank di Mr. Son sia uno di questi nuovi attori sul mercato. Io rispetto molto questa persona: è giovane e si sta concentrando su un business conservativo e non troppo diversificato. Potrebbe avere un grande successo già quest'anno e penso che dovrei incontrarlo, come sto incontrando altri imprenditori.

Cj: infine, Mr Shiina, quali sono le migliori opportunità per una azienda straniera che volesse entrare nel mercato giapponese?

Ts: Penso che il mercato del software sia ancora molto da sviluppare. L'hardware sta diventando difficile perché i margini si stanno riducendo. Il software, soprattutto con l'avvento di Windows 3.15 di Microsoft non ha ancora esaurito il proprio potenziale di marketing. La localizzazione del software non è difficile e in Giappone si è venduto praticamente solo dei wordprocessor negli ultimi 10-15 anni, mentre altri software avranno performance di vendita molto migliori di quelle attuali.

[traduzione a cura di Tn]

Il testo originale dell'intervista si trova all'indirizzo:

http://www.japaninc.com/cpj/magazine/issues/1994/aug94/08inter.html

Retro Code

L'ottimizzazione in BASIC

Il vecchio codice rivisto dall'alto di una evoluzione durata trent'anni.

La necessità aguzza l'ingegno si dice comunemente. Questo aforisma è sicuramente vero nel campo della programmazione dei calcolatori, dove spesso chi se ne occupa si trova difronte ad ostacoli non ancora affrontati da lui stesso o, per quanto di sua conoscenza, da altri.

La nostra moderna società dell'informazione ha di fatto ridotto drasticamente questo scenario, tanto che è praticamente impossibile che non si trovi oggi un esempio, un pezzo di codice, una "dritta" per superare quello che ci sembra arduo, in pochi spunti ben assestati. Lungi dal criticare l'attuale abbondanza di informazioni: una vera ricchezza senza ombra di dubbio, osservo che da un certo punto di vista questa "evoluzione collettiva" nell'ambito della programmazione ha un po' "addormentato" le persone che non trovano più il piacere della sfida e preferiscono frugare nella rete piuttosto che fra i loro neuroni alla ricerca di una soluzione belle e confezionata.

Quando i programmatori erano uomini (retorico, ma efficacie), era necessario riuscire a cavarsela con pochi spiccioli di informazioni, con l'esperienza e con grandi passaggi di test e ottimizzazione.

Nel BASIC, linguaggio interpretato per antonomasia, l'ottimizzazione del codice, sia per risparmio di memoria che di tempo di esecuzione, poteva avvalersi di pochi "sani" principi. Essi prendevano spunto dalla struttura elaborativa del linguaggio e, conoscendone i punti deboli, dettavano quelle regole cui si tentava di adeguare il proprio codice.

Nella ottimizzazione possiamo, come già accennato in precedenza, distinguere due ambiti: la dimensione del codice e la velocità di esecuzione. Spesso queste due esigenze sono in contrasto, nel senso che ad esempio un codice più "flat" è più prolisso ma permette una esecuzione più fluida, mentre il contenuìimento dello spazio obbliga magari alla scelta di strutture dati meno efficienti ma più contenute oppure moltiplica le chiamate a funzioni, con inevitabile overhead di tempo dovuto ai meccanismi di stack necessari al passaggio dei parametri alle funzioni stesse.

Ottimizzare lo spazio

Il BASIC è un linguaggio interpretato, pur tuttavia un minimo di ottimizzazione del codice è da sempre stata inserita in esso con la trasformazione delle parole chiave

in altrettanti "token" che occupano meno spazio (uno/due byte) e precompilano, per così dire, gli statement prima dell'esecuzione.

Per conservare la memoria si ricorre alle seguenti tecniche o, per meglio dire, attenzioni.

- inserire più statement sulla stessa riga. Quando questo è possibile una riga multipla risparmia spazio e velocizza certe operazioni; il simbolo che separa uno statement dal successivo è nella norma i due punti.

- rimuovere i commenti.

- usare di preferenza il GOTO piuttosto che il GOSUB. Il secondo è più flessibile e rende il codice più "elegante", ma introduce uno spreco di byte.

- usare variabili intere dove possibile. Questo è banale: l'immagazzinamento dei valori in memoria richiede più byte per una variabile floating point, peggio se in doppia precisione.

- usare poche parentesi nelle espressioni numeriche. A parte lo spazio per conservare il simbolo di parentesi, si innesca una ulteriore necessità di spazio quando l'interprete deve valutare l'espressione.

- ridurre il numero di variabili e array in un programma. Spesso una stessa variabile può essere usata per scopi diversi in punti diversi del codice.

- dimensionare gli array per lo stretto necessario, senza voler strafare. Ad esempio de un vettore richiede al massimo 88 elementi, perché dichiarare una variabile indicizzata di 100 elementi? Spesso i programmatori hanno queste sviste, forse perché usare una dimensione intera è più facile da portare avanti nel codice, cioè si ricorda meglio la dimensione. facciamo un esempio: se ho bisogno di un array di 23 righe per 88 colonne: A(23,88), potrei essere tentato di "esagerare" dichiarandolo: A(100, 100). Il fatto è che quando ci sia da fare un loop su di esso, magari molte volte nel corso del programma, viene più facile ricordare che l'array era di dimensione 100 piuttosto che 23x88!

- ridurre il numero di file aperti contemporaneamente; ogni file richiede un buffer di I/O ed è facile che esso richieda almeno 128 byte.

- usare lo statement DEF per dichiarare la tipizzazione delle variabili, invece che usare i caratteri trailing dopo il nome (ad esempio % per le variabili intere o # per quelle in doppia precisione); si risparmia un carattere ogni volta che la variabile viene nominata.

- per quegli interpreti che lo consentono l'uso della modularizzazione del codice è una vera manna dal cielo per il problema di occupazione di memoria. In effetti se si deve organizzare un programma con un menù cui seguono le relative funzionalità, è possibile avere un progrmma "master" che visualizzi il menù di scelta e che richiami

via "CHAIN" o equivalente, il relativo codice per la funzione desiderata.

Ottimizzare il tempo

Le indicazioni elencate nel paragrafo precedente sono spesso, anche se non sempre, in contrasto con la velocità di esecuzione del programma, oltre che in contrasto rispetto alla chiarezza del codice prodotto. basta pensare all'utilizzo di una stessa variabile in contesti diversi e alla mancanza di commenti, per capire che il codice diverrà ben presto difficile da manutenere.

Le ottimizzazioni del tempo di esecuzione passano anch'esse per buona parte delle indicazioni viste in precedenza. Ad esempio è evidente che l'uso indiscriminato di parentesi per raggruppare le espressioni numeriche è utile per chi legge il codice ma sovrabbondante per l'interprete.

Elenchiamo e discutiamo brevemente le indicazioni per minimizzare il tempo di esecuzione.

- usare variabili intere, soprattutto nei loop.

- dichiarare le variabili all'inizio del programma e definire per prime quelle usate più frequentemente. Infatti l'interprete si crea una sorta di tabella con nomi di variabili e corrispondente puntatore in memoria. tale tabella deve essere scorsa tutte le volte che una variabile viene chiamata in causa nel codice: se è all'inizio della tabella viene trovata prima.

- inserire il codice delle funzioni e subroutins più usate all'inizio del programma. Questo per lo stesso motivo rappresentato nel punto precedente. L'interprete BASIC a fronte di una chiamata a subroutine o anche ad un semplice GOTO, deve per forza di cose andare a scoprire il punto dove inizia il codice della funzione all'interno del programma. Esso, essendo non compilato, ha solo due possibilità: o si è creato, magari al momento del RUN una tabellina di corrispondenza fra numeri di riga e relativo puntatore di inizio riga in memoria, oppure scorre semplicemente il sorgente dall'inizio per fermarsi al punto voluto.

I primi BASIC seguivano decisamente la seconda strategia, mentre successivamente, quando la disponibilità di RAM ha dato maggiore respiro, si è cominciato ad utilizzare le tecniche di pre-compilazione per ottenere uno schema esecutivo maggiormente efficiente.

Quante volte abbiamo visto i sorgenti BASIC di un certo impegno cominciare con un salto ad una riga molto alta in numerazione, qualcosa come:

```
10 GOTO 6000
```

infatti le parti iniziali, che di solito non trovano occasione di essere ripetute nel corso dell'esecuzione, stanno benissimo lontane dalla cima, in modo che una ricerca di altro codice non debba passare obbligatoriamente ad esaminarle tutte le volte.

- usare variabili invece che costanti numeriche all'interno delle espressioni.

Ad esempio usare:

*A = R * PI*

dove PI = 3.14

piuttosto che:

*A = R * 3.14*

Bisogna ricordare infatti che quando fa i calcoli l'interprete deve necessariamente convertire la stringa numerica in una rappresentazione numerica interna che sia coerente con il calcolo richiesto e questo occupa sia spazio che tempo.

- Evitare i confronti. L'istruzione IF occupa spazio e tempo di esecuzione e qualche volta non è necessaria. Ad esempio:

```
10 IF ELABORA = 0
   THEN ELABORA = 1
```

Se la variabile ELABORA serve per discriminare l'avvenuta esecuzione di una parte di codice è meglio non interessarsi se essa valga zero prima di impostarla a 1, ma impostarla direttamente:

```
10 ELABORA = 1
```

E' molto meglio fare una assegnazione inutile che fare una IF inutile!

Conclusione

Le tecniche di ottimizzazione, almeno le più elementari, dovrebbero costituire bagaglio indispensabile per qualsiasi "coder". Maggiormente significative esse appaiono quando ci sia da fare i conti con le limitazioni delle macchine sulle quali opera il codice stesso. All'inizio dell'era home computer di limitazioni ce n'erano a volontà, ma proprio per questo nasceva e si sviluppava nell'esperienza di un programmatore la cognizione di doverne tenere conto di queste limitazioni ed operare di conseguenza.

[Tn]

Biblioteca

Le monografie vecchie e nuove che rappresentano una preziosa risorsa per chi ama il mondo dei computer in generale.

Scheda

Titolo:

Geek House

Sotto titolo: *10 Hardware Hacking Projects for Around Home*

Autore:

Barry Press and Marcia Press

Editore:

Wiley Publishing, Inc

Anno: 2005

Lingua: Inglese

ISBN: 0-7645-57956-8

Geek House



Geek House è uno di quei testi che si possono definire "guide pratiche". In particolare questo si rivolge ai tecnici informatici e/o multimediali, oltre a coloro che in generale amano giochicchiare con gadget di vario genere per automatizzare alcuni elementi della propria abitazione.

Il testo è scritto da un ingegnere e da sua moglie/compagna, evidentemente altrettanto appassionata del modding della propria abitazione.

Il testo è del 2005 e, pur rimanendo sostanzialmente attuale, soffre di una certa "vetustà" rispetto ad alcune soluzioni che sono ora alla portata con più semplici tools. Rimane valido invece l'idea di fondo dei progetti, alcuni davvero interessanti e utili, che potrebbero sia dare il via ad una realizzazione pratica a casa nostra, sia fornire qualche spunto per ottenere qualcosa di più adatto alle nostre esigenze.

Personalmente la domotica (così si chiama la scienza che studia l'automazione delle abitazioni) non è mai stata nei miei interessi primari. Costruito un sistema di antifurto non ho trovato altrettanto interessante cablare un sistema di misura di altri parametri come temperatura o per l'irrigazione automatica del giardino. Dal mio punto di vista è meglio annaffiarlo manualmente il giardino, almeno si ha l'occasione di rilassarsi staccando qualche mezzoretta da hobby più sedentari. Come trovo interessante ma non così essenziale disporre di diffusori per ascoltare una stessa fonte sonora in tutto l'appartamento, comunque de gustibus...

Il pubblico statunitense è sensibilmente più attratto dal fai da te, anche perché non ci sono così tanti vincoli e restrizioni su quello che le persone possono fare dentro e fuori la loro abitazione.

In ogni caso i progetti sono affrontati con il giusto grado di approfondimento con schemi che vanno dal

generale al particolare, con utili consigli e suggerimenti su come superare gli inevitabili problemi di funzionamento e/o di affidabilità delle realizzazioni.

[Sn]

Contents:

DIR

Le pagine dedicate al sistema operativo CP/M

Storia del CP/M by John Wigley

I mainframe della prima generazione, a parte le dimensioni e il costo, avevano una cosa in comune: poca memoria. Potevano memorizzare dati su nastri o dischi, ma la memoria "core" (quella che adesso chiamiamo RAM) era limitata nelle dimensioni.

Ricavare il massimo vantaggio da questa situazione ha significato sviluppare un sistema che assicurasse lo scambio dati fra memoria e unità magnetica in maniera sicura, puntuale ed ottimizzata. Oggeti residenti nella core memory vengono salvati su un'area temporanea del disco o nastro, sostituiti con nuovi oggetti che vengono mandati in esecuzione e successivamente richiamati quanto l'utilizzo lo richiede.

Molti lavori teorici si sono occupati di questo meccanismo per trovare metodi tali da aumentarne l'efficienza, lavori che sembrano dimenticati oggi. [ndr. L'articolo è stato scritto nel 1994, prima dell'avvento di sistemi operativi multitasking sui PC come OS/2, Windows 95/NT, etc...].

Le operazioni di pushing, getting, etc... necessarie al meccanismo di scambio delle informazioni rendono la cosa molto complicata da eseguire e pertanto è necessario un aiuto automatizzato, cioè un Sistema Operativo. Questo è esattamente quello che correda un mainframe: con il costo di una macchina di calcolo di quella classe, l'aggiunta di un piccolo effort per il Sistema Operativo è più che giustificato.

Non passò molto tempo che i sistemi operativi diventarono molto sofisticati e anche molto costosi. Caratteristiche non alla portata dei micro computer che nel frattempo si affacciavano sul mercato: ad esempio l'Altair 8080, equipaggiato con i micro Intel 8008 prima e 8080 dopo.

Il ciclo si ripeteva: computer molto primitivi e con limitate capacità di memoria che hanno la necessità di disporre di un sistema di gestione delle risorse.

E' impensabile adattare un sistema operativo per mainframe alle caratteristiche dei "giocattolini" basati su microprocessore. Un personaggio che si è posto il problema ed è ben deciso a trovare una soluzione si chiama Gary Kildall (ora a capo della Digital Research). Gary, utilizzando un sistema di sviluppo della Intel, mise a punto un sistema operativo adatto ai

processori 8080 e di conseguenza ai micro computer che ne utilizzavano il chip-set.

Anche altre persone tentarono la stessa strada, quello che ha di differente il sistema progettato da Kildall è che apparve subito attraente per gli utenti dei nuovi micro che erano principalmente tecnici e programmatori. Questi usano questi giocattoli per trasportare i programmi da loro scritti sul mainframe ed usare il CP/M di Gary è la strada più facile. La sigla CP/M è l'acronimo di Control Program for Microprocessor.

Tutti i computer lavorano con file: i programmi sono dei file, i dati sono file, etc... Una parte importante del lavoro che deve svolgere un sistema di elaborazione è scambiare file da canali di Input e Output. C'è necessità di disporre di quello che viene chiamato un BIOS (Basic Input Output System).

Tornando al 1975 come avveniva il colloquio con i computer? Attraverso terminali telescriventi dove la tastiera fungeva come input e la stampa su carta come terminale di output (non esistevano i monitor crt). Esiste nel sistema una unità chiamata CCP (Console Command Processor).

Il primo passo di Gary Kildall è stato costruire un sistema di I/O per una unità floppy da 8 pollici, periferica comune nei mainframe e disponibile potenzialmente come unità di storage per i microcomputer. Questa parte del sistema venne chiamata BDOS (Basic Disk Operating System).

Ognuna delle tre componenti: BDOS, CCP, BIOS, possono lavorare indipendentemente ma il passo successivo è stato quello di farle lavorare assieme. Ogni comando battuto sulla tastiera viene interpretato dal modulo CCP il quale attraverso il BIOS richiama il BDOS, carica il file corrispondente in memoria e lo esegue.

La parte di memoria riservata al caricamento dei programmi si chiama TPA (Transient Program Area).

Il sistema può dirsi completo, ma come farlo partire? Qui dobbiamo ricordarci qual'era la situazione della memoria nelle macchine dell'epoca 1975. La maggior parte di esse aveva appena 4 Kbyte, troppo pochi anche per un sistema spartano come il CP/M.

Il primo problema è stato quindi quello di dotare le macchine di un minimo di 16 K di memoria RAM. Infatti calcolando all'incirca 1,5 K per il BIOS, 0,5 K di CCP e 2,5 K di BDOS si supera la soglia della dotazione minima delle macchine dell'epoca, senza contare che servono aree di buffer e soprattutto l'area TPA per eseguiri i programmi caricati. Da ricordare inoltre che per usare il CP/M bisogna avere una unità disco: senza floppy nessun CP/M!

Una bella immagine di Gary Kildall, ideatore del CP/M e fondatore della Digital Research.



I primi sistemi commerciali ad usare il CP/M in maniera "automatica" sono stati il TRS80/System80 (conosciuto anche con il nome di The Sourcerer). L'hardware di questo sistema fu modificato in modo che all'accensione venisse caricato in memoria ed eseguito il contenuto del primo settore del floppy. Così allo start-up il sistema carica il CCP, il BDOS e il BIOS nella parte alta della memoria, tutta la zona "bassa" rimane a disposizione per la TPA.

Il vantaggio di questa configurazione è che il CP/M funziona con qualsiasi dotazione di RAM trovata sul computer ospite (da 16K fino a 64K).

La seconda caratteristica da conoscere è come viene gestita l'organizzazione dei file sull'unità magnetica, il cosidetto "file system". Per il CP/M un file viene individuato da un nome di otto caratteri come massima lunghezza e da un tipo che è una estensione di tre caratteri. Così ad esempio "BIGFILE.BAS" indica il file "BIGFILE" di tipo "BAS" (un sorgente BASIC per convenzione).

I file sono immagazzinati sul disco in blocchi da 128 byte ciascuno. Non ha importanza l'ordine di memorizzazione perché una struttura tabellare (File Control Block) si occupa di tenere traccia di questo. Per effetto delle scelte progettuali, ogni file può essere al più grande fino a 256 Kbyte.

Queste le basi del primo CP/M fino alla versione 1.4; successivamente è uscita la versione 2 che ha portato questo limite a 8 Megabyte e soprattutto ha aggiunto la possibilità di lock dei file. L'annunciata versione 3, chiamata anche CP/M Plus, incrementa notevolmente tutte le possibilità tecniche, aggiunge password e protezioni all'uso dei file e ottimizza l'accesso ai dischi grazie ad un nuovo meccanismo di caching della memoria.

[Traduzione a cura di Tn]

Bibliografia

L'articolo originale è apparso su Bits&Bytes NZ edition, Agosto 1983, pag. 33

La storia di Gary Kildall e del CP/M è stata raccontata in un video documentario della serie Computer Cronacles e si può trovare on-line all'indirizzo:

http://www.archive.org/details/GaryKild



THE
CP/M
COMPATIBLE
SOFTWARE
CATALOG
DIGITAL RESEARCH

Retro Riviste

La rassegna dell'editoria specializzata dai primi anni '80 ad oggi

Elettronica hobby

Scheda

Titolo:
Elettronica Hobby

Sottotitolo:
Mensile di elettronica e hardware per microcomputer

Editore:
Gruppo Editoriale Jackson

Lingua:
Italiano
Prezzo:
3.500 Lire

Primo numero:
1985



La Jackson una ne fa e cento ne pensa, è proprio il caso di dire. Non si contano le riviste e opuscoli vari messi in cantiere da questo editore. Qualcuna ne ha azzeccata, qualche altra proprio no, ma ci ha provato continuamente, forte della sua dimensione aziendale, della possibilità di riciclare il personale fra una redazione e l'altra e del possesso dei diritti di pubblicazione di numerose riviste statunitensi che non le hanno mai fatto mancare la materia prima per le pubblicazioni.

L'esagerazione, perché di questo si deve parlare, ha portato l'azienda a voler coprire ogni possibile settore di mercato con poca attenzione alla qualità. Alcune sue pubblicazioni, soprattutto monografiche, sono emblematiche di come NON si dovrebbe produrre una documentazione tecnica.

Come altre consorelle dello stesso editore, anche EH è stata soggetta a revisioni e la ritroviamo nel 1987 con un diverso logo e sottotitolatura e con un residuo di progetti dedicati al microcomputer. Forse, anzi è molto probabile, che nel frattempo fosse nata qualche altra testata dello stesso editore orientata maggiormente verso l'elettronica digitale o forse (siamo ormai nel 1987) l'editore avesse percepito che la stagione del computer autocostruito si stava esaurendo.

Elettronica Hobby è la tipica pubblicazione Jackson. Essa non brilla per qualità o spessore del contenuto, ma vivacchia fra qualche progetto discreto e robaccia improponibile.

La parte "micro" è però sempre interessante, proprio per la possibilità che dicevamo di poter pescare dal bacino dell'editoria specializzata anglosassone.

Ne posseggo solo due numeri



(1985 e 1987) e quindi posso basarmi solo su questa limitata campionatura. Comunque troviamo nel primo fascicolo una interessante interfaccia Centronics per C64, completa di software per l'utilizzo. Niente di eccezionale: un interfacciamento della User Port con un "driver" basato su poke.

Sul secondo fascicolo la "solita" penna ottica, sempre per C64. Una periferica questa, la penna ottica, che ha spopolato parecchio all'epoca. In qualche misura era curiosa l'idea di poter interagire direttamente con il video del computer, piuttosto che con la tastiera, obiettivamente scomoda nelle applicazioni di grafica. Per fortuna che poi la Xerox ha inventato il mouse e che la Apple l'ha imposto, perché usare la penna ottica è quanto di più scomodo si possa pensare: non è così precisa come sarebbe necessario e soprattutto ci si stanca da morire a tenere il braccio alzato a livello dello schermo.

ù

In conclusione una rivista che salviamo per quel nostro istinto di conservazione ma che ha ben poco da dire sul fronte micro computer.

[Sn]



SOMMARIO

pag. 24 IL C 64 COME COMBINATORE TELEFONICO

pag. 16 TELEFONO HANDS-FREE

JACKSON

Anno 5 • N. 31 MARZO 87

Il racconto

Rubrica curata da [Bs]

La Megaditta - La Fine

Storie di vita dove i computer (soprattutto retro computer) c'entrano in qualche modo.

Ultimamente è stata organizzata la solita cena annuale della megaditta. Dal momento che paga il megadirettore megagalattico (non ci crederete ma è vero), corriamo sempre tutti quanti. Infatti erano più di due anni (dall'ultima volta che la megaditta era in attivo, presumibilmente :-) che non si faceva.

Da tempo ormai ho cambiato lavoro, ma non per questo ho dimenticato i vecchi amici, compagni di tante avventure che ho cercato di raccontare in questa specie di diario.

Ebbene, dopo averci per mesi promesso un pranzo luculliano, una grande abbuffata in stile "il ricco Epulone", un banchetto pantagruelico che avrebbe fatto rabbrividire i pranzi da quaranta portate dello scia' di Persia (ma non ricordo quale scia'), veniamo una quindicina di noi, praticamente tutta la megaditta, più due estranei, i responsabili della ditta XXX, evidentemente i clienti di turno da blandire. Ricordo qualche lavoro tentato per questa ditta e in particolare ricordo che una volta ci stavano creando problemi di svenimento perché eravamo andati ad una riunione con loro alle 12:30, praticamente ora di pranzo, ed avevamo una fame da lupi, e quel posto sembrava la sagra delle tentazioni della gola...

Ebbene, tornando alla cena conviviale, dopo averci promesso per mesi un pranzo da "meglio panza che schiatta, che roba che resta", andiamo avanti con una serie di "assaggini" (il diminutivo era tutt'altro che fuori luogo) che alla fine ci lasciarono con una fame praticamente doppia (nota: quando io sono sotto tensione, io mangio più di un branco di piranha).

Avremmo divorato le gambe dei tavoli, se solo avessimo avuto un po' di sale. Beh, pazienza, non si può avere tutto dalla ditta, pardon, dalla vita. I due clienti, in "cena di lavoro", avevano gli occhi fuori dalle orbite dalla fame, e trattenevano a stento imprecazioni in aramaico ed altre quarantadue lingue da tempo scomparse (o forse erano solo grugniti di disapprovazione quelli che emettevano).

Bene, prima di uscire da quell'ameno localino, dove c'era anche una banda di tedesconi al limite dell'ubriachezza molesta - e che noi in ogni caso invidiavamo perché si erano abbuffati come maiali di quelli che vengono nutriti con tante cose buone perché dopo ci devono fare dei prosciutti di pri-

ma qualità - vi racconterò di una piccola discussione che ho avuto con XXX, il programmatore che lavorava prima di me nel 19XX alla megaditta e che io sostituii quando lui si licenziò (o si fece licenziare? boh? mai capito).

Il tizio era iscritto a informatica e se la cavava pure abbastanza bene. Abbandonò informatica e si mise a lavorare da solo. Tempo fa si iscrisse a teologia a Roma. Gli chiesi come mai avesse abbandonato informatica, pensavo che non avesse la forza di studiare. Niente, si era scocciato, voleva guadagnare. Fin qui niente di clamoroso. Il clamoroso viene quando si è iscritto a Teologia. Gli chiedo come era andata. "Beh, ho rifiutato 28/30 all'esame di XXX, volevo 30". Poi abbandonò anche quella facoltà, per sposarsi. Conobbe una ragazza e dopo diciannove giorni si sono sposati!

Ebbene, gli mancava pure la patente. A 25 anni uno senza patente è proprio una frana. Infatti stavamo tornando a casa dopo la cena di cui sopra, eravamo in una 126 più scassata di quanto un povero napoletano di provincia possa immaginare. Pioveva a dirotto.

Dietro sedevano due altre persone, XXX (è il cognome, occhio), e l'ingegner XXX (un gran XXX!), probabilmente già in pensione ma che ancora ha forza, voglia e tempo di fare il consulente per la megaditta.

Davanti, alla guida, il figlio (estasiato dai miei complimenti perché lui era riuscito a vedere il Monsters of Rock e io no :-), gli avrei dato un 15-16 anni, però guidava lui, beh, che dire? Se la cavava. Ed ovviamente io.

NON SI VEDEVA NIENTE!!

Una cosa incredibile, pioveva manco fosse un nuovo diluvio universale. Il tizio per fortuna aveva buonsenso e andava come una lumaca. Improvvisamente si ferma (non di colpo altrimenti chissà dove saremmo "scivolati", pareva di stare su un hovercraft su un fiume in piena, tant'acqua che c'era).

Un decimo di secondo dopo si sente un PATACRACK! Un ciclomotore "Sì", con a bordo padre e figlio (entrambi senza casco), aveva tentato di evitare la 126 e metteva la "freccia" (sporgeva il braccio) per girare al semaforo un paio di centinaia di metri più avanti, ed invece scivolarono rovinosamente (forse perché il braccio aveva avanzato troppo il baricentro) - a quella velocità avremmo raggiunto entrambi il semaforo per il mattino dopo, quindi vi lascio immaginare la tempestività dell'azione.

I tizi si rialzano e si accostano sotto un portico a contare i danni, e mentre noi ripartiamo, ci fanno segno di fermarci. Due carabinieri intervengono a vedere cos'era successo.

La 126 non era stata neppure toccata, al "Sì" era rimasto un pedale storto. I carabinieri perdono qualche attimo coi due tizi in motorino, che dopo un minuto di imprecazioni ripartono (non erano riusciti a darci la colpa, e poi non si può dire "il motorino non funziona" se il pedale

è un po' storto :-).

I carabinieri fanno, con voce glaciale: "documenti". In quell'attimo l'ingegner XXX cambia colore.

SENZA DOCUMENTI!!!!!!!!!!!!!!!! La 126 era sprovvista di documenti e l'ingegneraccio vecchietto aveva solo spiegazioni filosofiche (ma dubito che i caramba avrebbero capito la spiegazione "ma Dio ce l'ha data così, senza documenti, non ci avevano mai fermato" :-).

Il figlio, alla guida, pure senza patente né documenti. Il padre pure senza documenti, se si esclude una carta d'identità non valida (non si capisce se scaduta, se mancante di qualcosa, se strappata etc). E pure XXX senza patente. Infine il sottoscritto.

Tiro fuori al momento giusto la mia patente giusta, e quel carabiniere me la fa cascare a terra in una pozzanghera (nella "lózza"!!) rovinandomela irrimediabilmente (ce l'ho sempre avuta senza custodia, da 5 anni pare nuova, nuovissima, neppure un millimetro quadrato sporco o piegato o rovinato, ed un carabiniere sotto una pioggia battente che me la lascia cadere nella me... nella melma!).

I carabinieri invece di farci passare una notte al fresco come meritavamo, incredibilmente se ne escono con un "potete andare, basta che guidi lui" indicando ovviamente me, detto "l'Asso del Raccordo Anulare", "il Pantofola D'Oro dell'Acceleratore", "il Fulmine della Via Emilia". Ovviamente vi ho risparmiato una lunghissima discussione a base di "perché lui non ha la patente" e "perché la macchina è senza documenti": ci avranno ripetuto quelle domande una cinquantina di volte a testa, ogni volta con una risposta diversa e sempre meno plausibile. Alla fine devono essersi scocciati e ci hanno lasciato andare (posso negare l'intera storia: forse hanno riconosciuto personalmente uno degli altri tre che erano in macchina con me. O forse, chissà...).

Dopo 100 metri, alla prima a destra, ho accostato e ho ceduto la guida ad XXX perché io con le 126 proprio non ci sono abituato. Sapete com'è, avendo guidato sempre macchine grosse e potenti (Volvo 244, Prisma TD, etc) non sono abituato a queste carriole!

Vi risparmio anche la lunga sequela sull'acqua, ne ho vista più io quella sera che tutti voi durante un decennio di stagioni delle piogge.

Un particolare però della serata non posso risparmiarvelo. Il megaboss che diceva "Io vorrei significare questa serata..." - segue discorso fitto di buone intenzioni, che comincia a perdersi, a divincolarsi, a cambiare direzione, per finire in uno schema elettrico di un nuovo apparato, alla ricerca del blocco appunti perduto, mentre quei tedesconi si abbuffavano come pozzi senza fondo e noi contavamo gli otto o dieci gnocchi a testa (dieci ai più fortunati).

Il giorno dopo, nella megaditta decimata (dai mal di pancia per aver

mangiato troppo la sera prima? ;-), c'è stato un lunghissimo battibecco tra il sottoscritto e la segretaria "amministrativa" sul numero di giorni e di ore lavorate. Poi quando sono riuscito a dirle che non mi servivano soldi stavolta, che non stavo a batter cassa, ho visto un sorriso di quelli da manifesto elettorale americano e si è placata...!

Il giorno dopo, caso forse unico di tutte le volte che ho lavorato da Roma in su, mi liberai ad ora di pranzo ed andai a trovare Caterina.

Dopo non molto tempo, la Mega-Ditta chiuse i battenti, ed io cominciai a lavorare da libero professionista a sud di Roma. Ma questo ve lo racconto nel prossimo volume...

[Mm]

[Nota del curatore Besdelsec]

Le storie della Magaditta terminano effettivamente con questo ultimo piccolo episodio. Maurizio, protagonista e autore di questo diario, ci ha fatto rivivere negli esilaranti e a volte assurde situazioni, una atmosfera che forse non sarà parsa estranea a molti di noi.

Una trentina di anni fa questa era più o meno la situazione di chi si apprestava a cimentarsi con la professione dell'informatico: ditte fatiscenti sempre sull'orlo del fallimento, richieste assurde e orari altrettanto assurdi, paghe a volte buone ma mai certe, etc...

Di buono c'erano le invenzioni che quotidianamente eri chiamato a tirare fuori dal cappello a cilindro, visto che sistemi (i PC) erano poco standardizzati e i linguaggi poco ricchi di features, lontani mille miglia dalle soluzioni precotte di oggi.

Le varie megaditte, cui l'Italia era piena e forse lo è ancora, hanno però avuto il pregio di permettere la crescita di tanti giovani che si sono poi trasformati in professionisti e hanno fatto fare un salto di qualità tecnica all'intero paese. Se avessimo aspettato gli ingegni di quei sedicenti ingegneri informatici, avezzi al camice bianco e molto meno al saldatore, forse saremo ancora a terminale teletype e alle pizze di nastro da un quarto di pollice.

Come eravamo...

Storia dei videogiochi (parte 3)

La storia dei sistemi e degli uomini che hanno creato un mondo nuovo.

Abbiamo lasciato i videogames nella puntata precedente al momento della transazione verso la grafica, cioè in quel particolare momento evolutivo che vede l'aspetto grafico ad assumere importanza predominante.

Altri generi si adeguano e fanno della grafica un punto di forza. Ad esempio il genere avventure che si trasforma in una ricerca di indizi e soluzioni di enigmi in prima persona. Alone in the Dark è il capostipite del genere Horror Adventure nel quale il giocatore vive una esperienza meno adrenalinica di quella possibile in Doom, ma con un vero coinvolgimento emotivo fino a sfiorare la soglia della paura.

Nella ricostruzione di un environment spesso viene citata la grafica come motore trascurando l'aspetto suono che invece secondo me non è stato inferiore. L'evoluzione delle interfacce sonore, grazie all'egregio lavoro di sviluppo portato avanti dalla Sound Blaster e l'inserimento nella colonna sonora di effetti "circolari", hanno contribuito come la grafica a rendere l'esperienza del giocatore sempre più reale. Un conto è vedere arrivare sul davanti un mostro sferragliante pieno delle peggiori intenzioni nei nostri confronti ma altra cosa è sentire "dietro" un terrificante respiro del nemico che ci è giunto alle spalle.

Alone in the Dark: il primo titolo a fare della grafica un must.



Il primo videogioco che ha avuto bisogno di un intero supporto cd-rom e che nelle release successive non ha fatto che aumentarne il numero è "Myst". La grafica di Myst è considerata un capolavoro vero e proprio e l'atmosfera in cui si viene immersi vale tutto il prezzo pagato per acquistarne una confezione.

C'è perfino chi sostiene che Myst sia stato la killer application per la diffusione del cd-rom come supporto di storage per il PC.

In generale tutti i giochi di simulazione, quelli di guida aerea in primis, hanno trovato nell'evoluzione grafica un motivo di sviluppo e di sopravvivenza. Flight Simulator della Microsoft è forse il titolo più noto del genere, ma le simulazioni di battaglie aeree o anche spa-

ziali (la serie Wing Commander, per esempio), non sono state da meno.

A metà degli anni '90 si assiste anche alla vera integrazione gioco-film con una connivenza dei due mezzi sempre più indistinguibile. Da una parte gli effetti speciali usati nelle pellicole di Hollywood fanno assomigliare il film ad un gioco su PC, dall'altra i giochi che ospitano sequenze animate registrate da veri attori. La serie Wing Commander, già citata, con le sue otto se non più incarnazioni, ha fatto la fortuna della Origin System. La serie ideata e prodotta da Chris Roberts, è perfino arrivata a trasformarsi in una pellicola, anche se con meno fortuna di quanto il suo creatore aveva certo sperato.

Altro fenomeno di "ingegnering inverso" è stato il gioco Lara Croft, diventato famoso anche per la trasposizione a video (Angiolina Jolie come protagonista).

I primi anni '90 vedono anche la nascita di un nuovo genere: la simulazione. L'idea della simulazione c'è sempre stata nei programmi per calcolatore, ma solo una certa classe di processori permette il controllo di un numero di parametri sufficiente per rendere il tutto credibile dal punto di vista grafico e di numerosità degli oggetti controllabili. Il rappresentante principe di questa categoria è sicuramente Sim City. L'idea è quella di simulare un mondo più o meno reale con tutte le difficoltà di far "quadrare i conti" e sviluppare un'azienda fino ad una intera civiltà.

L'ultimo rappresentante di questo genere è The Sims mentre già si intravvede il futuro: l'avatar e il mondo simulato di Second Life.

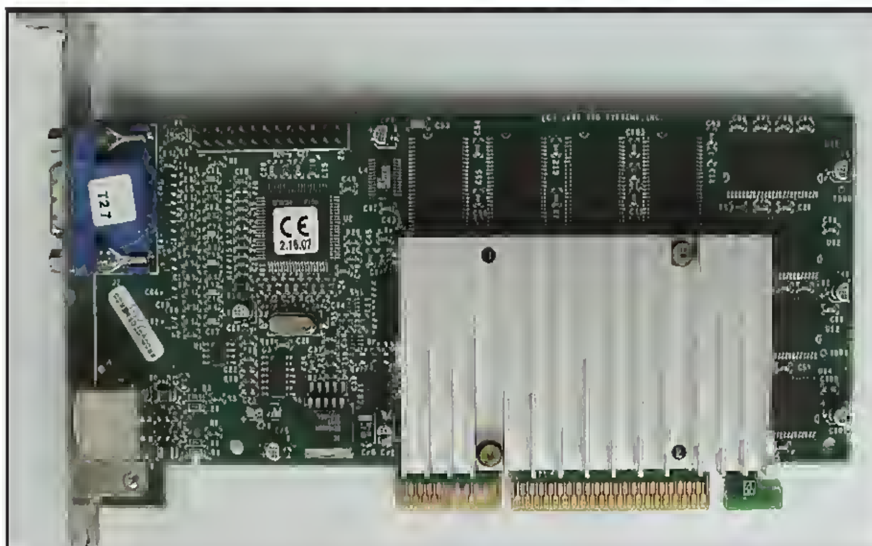
Tornando agli anni '90 la fame di cicli di clock non poteva non generare una specializzazione nella gestione della grafica. E' di questa epoca la nascita delle schede acelleratrici (3DFX) che affiancano la normale scheda video ma si occupano della maggior parte del la-



Myst, dove la grafica raggiunge livelli poetici.

La saga di Wing Commander si trasforma in cinema





La scheda accelleratrice 3Dfx.

voro "sporco", sgravandone il processore centrale.

La ricerca dell'ultimo FpS (Frame per Second) ha raggiunto livelli di parossismo già attorno al 1994: due schede che lavorano in parallelo e che da sole assorbono più corrente di tutti gli altri componenti del PC messi assieme! Non staremo esagerando?

Il fatto che spesso si compri il PC solo per giocare, unita alla difficoltà di gestione di un personal, ha dato un po' di fiato all'industria delle console che è tornata alla grande con oggetti tipo PS2 e la ancora più innovativa PS3, per non parlare dell'altra trovata della Nintendo (la Wii) con quella sua idea di interagire con il gioco muovendosi fisicamente. Non sappiamo ancora l'impatto sociale di questa innovazione ma anche qui in realtà sono cose già viste, certo riproposte in chiave moderna con tutta l'innovazione tecnologica possibile.

L'innovazione "clamorosa" è meno presente negli anni dal 2000 in poi. La grafica incrementa costantemente le proprie performance e il tutto diventa forse anche troppo realistico, viceversa si vede come nuove idee stiano cercando di sfruttare aspetti meno tecnici ma più emozionali. Second Life appartiene a questa categoria; la sua grafica (diciamolo) fa schifo, la possibilità di avere a video molti oggetti animati contemporaneamente ha fatto grandi passi indietro, ma SL aggiunge un effetto "presenza" che fino alla sua introduzione si era visto solo come esperimento isolato. Ora il mio "avatar" me lo costruisco come voglio (Mods) e quando partecipo ad un evento sono "io" (per modo di dire).

La diffusione dei servizi in rete Internet ha coniato l'assunto "in Internet nessuno è quello che dice di essere", il che è la versione moderna della Pirandelliana filosofia del "così è se vi pare" o meglio di "Uno, nessuno e centomila".

Le riviste

Uno dei fenomeni collaterali del successo dei videogame è sicuramente la nascita delle riviste periodiche dedicate all'argomento. In un primo tempo esse si sono mascherate sotto una patina di serietà prevedendo il supporto all'utenza, che si credeva capace di trasformarsi in massa in programmatori, rispetto all'uso del personal appena ricevuto in regalo. Successivamente sono stati rotti gli indugi e prima negli States, poi in Europa, le testate inequivocabilmente le-

gate all'intrattenimento elettronico sono state fondate e sembra non abbiano mai conosciuto crisi. Forse solo la proliferazione dei siti Internet sull'argomento potrebbero costituire per esse una minaccia ma l'abbondanza lo si sa è l'antitesi della chiarezza e non è detto che l'utente non preferisca sfogliarsi con calma una rivista cartacea piuttosto che ciccare freneticamente da un sito all'altro.

Le riviste hanno avuto un grande aiuto anche nella possibilità di distribuire i supporti con i vari trial dei giochi e ultimamente anche dei giochi completi, seppur "vecchiotti". La presenza di questi gadget, se ben supportata da articoli introduttivi e di spiegazione di trucchi e accortezze per ricavare il miglio godimento dal titolo, costituisce un add-on irrinunciabile per la maggior parte degli utenti, almeno a giudicare dai sondaggi.



The Sims è un mondo virtuale molto vicino a quello reale.

Conclusione.

E' tutto un gioco o ci stiamo avvicinando pericolosamente alla soglia della realtà? Probabilmente ci sono in giro fenomeni di sdoppiamento della personalità dovuti all'eccessiva frequentazione di Second Life e dall'identificazione attraverso l'avatar di noi stessi, insoddisfatti del nostro vivere quotidiano.

Forse sarebbe meglio fare un passo indietro e ritornare a giocare a scacchi…

[Traduzione a cura di L2]

Apple Club

Tutti i linguaggi dell'Apple (9)

La mela come paradigma della programmazione

```
+#
+# Welcome to the AZTEC C65 APPLE II
+#  Compiler System
+#
-?ls


DISK VOLUME 254

 A 002 HELLO
 B 059 SHELL
*T 002 .PROFILE
*B 019 CONFIG
*B 003 TABSET
*B 006 LOEV
*B 138 SHELLDEV.ARC
*B 070 VEOSRC.ARC
*B 063 PROGSRC.ARC
*B 018 OVERLAY.ARC
*B 047 UTIL.ARC
*B 035 NATIVE.ARC
-?
```

Introduzione

Atzec C è un compilatore prodotto dalla Manx Software Systems e disponibile per le più diffuse piattaforme in voga negli anni '80, MS-DOS compreso. Non si tratta di un semplice compilatore ma di un vero e proprio ambiente di sviluppo, addirittura cross-platform, che nel caso dei sistemi home rappresenta una occasione rara per disporre di un software di questa classe e di un linguaggio che negli anni '90 ha spopolato.

Nel caso specifico dell'Apple II ne esistono due implementazioni: per DOS 3.3 e per PRODOS. Tramite emulatori e utilizzando la versione che funziona in Windows, è data ancora oggi la possibilità agli appassionati (che sono tanti) di realizzare buoni programmi e trasferirli poi in ambiente di emulazione o sui sistemi obsoleti addirittura.

Per quanto riguarda le condizioni d'uso, la Manx (o meglio il titolare dei diritti Harry Suckow) ne permette il libero uso sotto una sorta di agreement "fair", il che significa che potete usarlo liberamente per produrre software per uso personale ma non per venderlo in giro.

Nelle bibliografia forniamo l'indirizzo di un ottimo sito dove trovare documentazione e altro.

Il linguaggio C trasforma l'Apple II in una workstation di sviluppo.

```
VED 2.1 - hit return to continue

i           insert text
escape      end insert
r           replace character
x           delete characters
dd          delete lines
yy          yank lines
yd          yank and delete lines
p           put yanked lines
-           previous line
return      next line
0/$         beginning/end of line
g           go to line
h/m/l       top/middle/bottom of screen
z           redraw the screen
/string     search for string
n           repeat last search
:q          quit
:w          write file
:r          read file
:e          edit new file
:f          file stats
```

Help del programma VED, l'editor di testi dell'ambiente.

Breve storia

La Manx Software Systems of Shrewsbury nel New Jersey ha cominciato a produrre un compilatore C cross-platform negli anni '80, puntando all'ambiente di sviluppo professionale.

Gli sviluppatori principali furono: Harry Suckow, Thomas Fenwick, e James Goodnow. Suckow fu uno di quei personaggi visionari che sfornavano idee di business a getto continuo, fondavano proprie società (che poi vendevano o chiudevano per fallimento) e riuscivano a catalizzare attorno al proprio entusiasmo un ambiente creativo per i tecnici e programmatori dell'epoca. Suckow nella Manx Software era l'anima commerciale mentre agli altri due rimaneva il compito duro di sviluppare fron-end e back-end del prodotto.

Uno dei successi del prodotto fu l'avere messo a disposizione in un compilatore una libreria per l'aritmetica floating point per codice Z80, presto esteso al 6502 nella versione Apple. L'Atzec C rimase il riferimento del mercato home per questo genere di prodotti almeno fino al 1989, quando altre società e in particolare Microsoft cominciarono ad interessarsene offrendo alternative soprattutto per le macchine di punta: MAC e PC-IBM.

L'Atzec C continuò ad essere offerto sul mercato anche negli anni '90 ma la società perdeva rapidamente profitti e perse pure il treno degli ambienti di sviluppo embedded.

Alla fine la società fu costretta a chiudere ma l'ambiente di sviluppo è rimasto tuttora vivo per la gioia di moltissimi appassionati che trovano divertente sviluppare software per le vecchie piattaforme.

Il prodotto

Come si diceva l'Atzec C non è un semplice compilatore ma un vero e proprio ambiente multipiattaforma. Esso si compone sostanzialmente di due parti distinte: il front office, cioè la shell con le utilities di sviluppo e il compilatore vero e proprio.

La shell, come viene chiamata, è un ambiente Unix-like nel senso che i comandi sono quelli che conosciamo: cp per copiare un file, mv per rinominarlo, rm per cancellarlo, e così via.

Il compilatore è un sofisticato as-

```
-?c65 test.c
C II Vers. 1.05i 6502
(C) 1982, 1983 by Manx Software Systems
test.c:line 10: error 46, ''
1 errors
```

Prima compilazione e... primo errore di sintassi :-(

semblatore in grado di produrre sia codice nativo (nel caso dell'Apple per il 6502) ma anche un pseudo-code interpretabile a run-time. Quest'ultimo ha il vantaggio di essere portabile e di essere più contenuto nelle dimensioni ma ovviamente richiede la presenza delle librerie di run-time sulla macchina dove viene eseguito.

I due tipi di eseguibile (nativo e interpretato) possono coesistere nello stesso progetto per una ottimale distribuzione della elaborazione.

La versione per Apple II

La versione dell'Atzec C per il sistema Apple II di Apple Computer è distribuito su una serie di nove floppy che conservano la formattazione originale DOS o PRODOS, divisi in Shell, Compilatore e utilities. Il manuale cartaceo è abbastanza ricco ed esaustivo e descrive nel dettaglio tutti i comandi e le funzioni di libreria utilizzabili nel codice.

Fra i supporti, nella versione DOS 3.3 che abbiamo scelto, troviamo curiosamente due floppy 1: uno bootabile e l'altro no e un floppy siglato con il numero 8 che di fatto è vuoto e serve per farsi una copia di backup del boot originale. L'Atzec C customizza le tracce del floppy riservate normalmente al DOS (le prime due) e pertanto è necessario una apposita utility per la formattazione o più comodamente clonare il floppy fornito blank per avere i supporti necessari ai propri progetti.

Al boot l'utente si trova proiettato in un mondo affatto sconosciuto: la shell Unix per l'appunto e pertanto un utilizzatore Apple "classico" necessita di qualche ora di affiatamento prima di muoversi con agilità all'interno dell'ambiente. Anche per coloro che magari Unix lo praticano (o i pochi che lo praticavano negli anni '80) è consigliata una esplorazione attenta delle possibilità date dal programma. Non deve essere dimenticato che siamo su una piattaforma per forza di cose "ristretta" rispetto alle lussuose workstation Unix e non si può pretendere di avere tutto disponibile.

Le richieste hardware sono di un sistema a 80 colonne (consigliata), dell'espansione massima di RAM (64 Kb), dell'estensione per i caratteri minuscoli per le macchine più vecchie e di due drive floppy (si può lavorare anche con un floppy solo ma è scomodissimo).

I compilatori sono due: C65 produce il codice nativo per il processore e CCI produce invece lo pseudo code da interpretare a run-time. E' presente anche un assembler AS65 per il processore Apple, usabile anche stand-alone, ma ovviamente trova il suo uso ottimale

all'interno del processo di compilazione dei sorgenti scritti in C.

I comandi della shell vanno fornito in minuscolo, pena errore di interpretazione. Ad esempio:

```
ls
```

stampa a video la directory corrente (cioè il contenuto del floppy). Le due unità floppy sono individuate dai nome "d1" e "d2", così che:

```
ls d2
```

lista il contenuto del floppy nella seconda unità.

E' bene prendere confidenza subito con i caratteri di controllo che risultano maggiormente utili:

CTRL-C per interrompere qualsiasi esecuzione e tornare alla shell

CTRL-X per cancellare la riga corrente

CTRL-S per fermare e far ripartire lo scroll video

In accordo con le macchine Unix, esiste anche il CTRL-H per cancellare il carattere a sinistra del cursore, stessa funzione che viene fatta dal tasto "freccia a sinistra".

Primo approccio

A questo punto ci è venuta voglia di compilare il nostro primo programma in C su una macchina Apple II. Ci serve un editor per il sorgente. Il prodotto ha a corredo

```
-?c65 test.c
C II Vers. 1.05i 6502
(C) 1982, 1983 by Manx Software Systems
Aztec 6502 Assembler - Version 1.03
-?
```

Questa volta è andata: compilazione e assemblaggio sono stati portati a termine.

l'editor VED, che ha gli stessi comandi (o almeno una parte di essi) del famoso "vi" UNIX.

Il nostro primo programma è molto banale, poco più di un "hello world!":

```
#include <stdio.h>
main(void)
{
register int i;
printf("begin of program/n");
for(i=1; i<10000; i++);
printf("end of program");
}
```

La prima compilazione, chiamata con il comando "C65 test.c" è andata buca per via di un errore di sintassi: dimenticata la parentesi graffa chiusa della funzione main(). Poco male, corretto il problema e rilanciato abbiamo aspettato circa un minuto per compilazione e assembler.

I tempi possono sembrare eccessivi, anche data l'esiguità dell'esempio, ma non va dimenticato che il processo si svolge con il supporto dei floppy e quindi bisogna mettere in conto i tempi di accesso di questa periferica.

A questo punto sul floppy ci troviamo il file "test.rel" che contiene il codice rilocabile da linkare.

Il programma linker fornito dalla Atzec si chiama LN e sta sul floppy numero 5, denominato anche Li-

```
C II Vers. 1.85i 6502
(C) 1902, 1903 by Manx Software Systems
Aztec 6502 Assembler - Version 1.03
-?ln test.rel sh65.lib
C Linker Vers. 1.051
0ase: 0000   Code: 13de  Data: 00a4  Total: 001406
-?test
begin of program
1 2 3 4 5 6 7 0 9 10 11 12 13 14 15 16 17 10 19 20 21 22 23 24 25 26 27 20 29 30
 31 32 33 34 35 36 37 38 39 48 41 42 43 44 45 46 47 48 49 58 51 52 53 54 55 56 5
7 50 59 60 61 62 63 64 65 66 67 60 69 70 71 72 73 74 75 76 77 70 79 00 01 02 03
04 05 06 07 00 09 90 91 92 93 94 95 96 97 90 99 100 101 102 103 104 105 106 107
100 109 110 111 112 113 114 115 116 117 110 119 120 121 122 123 124 125 126 127
120 129 130 131 132 133 134 135 136 137 130 139 140 141 142 143 144 145 146 147
140 149 150 151 152 153 154 155 156 157 150 159 160 161 162 163 164 165 166 167
160 169 170 171 172 173 174 175 176 177 170 179 100 101 102 103 104 105 106 107
188 189 198 191 192 193 194 195 196 197 198 199 288 281 282 283 284 285 286 287
200 209 210 211 212 213 214 215 216 217 210 219 220 221 222 223 224 225 226 227
220 229 230 231 232 233 234 235 236 237 230 239 240 241 242 243 244 245 246 247
248 249 250 251 252 253 254 255 256 257 250 259 260 261 262 263 264 265 266 267
260 269 270 271 272 273 274 275 276 277 270 279 280 281 202 203 204 205 286 287
200 209 290 291 292 293 294 295 296 297 290 299 300 301 302 303 304 305 306 307
388 389 318 311 312 313 314 315 316 317 318 319 328 321 322 323 324 325 326 327
320 329 330 331 332 333 334 335 336 337 330 339 340 341 342 343
```

Ecco il nostro programma in esecuzione.

brary Disk. Per il nostro esempio, davvero semplice nella sua formulazione, bastano pochi secondi per generare il codice eseguibile con il comando:

```
LN TEST.REL SH65.LIB
```

Ed ottenere un file TEST sul disco, pronto per l'esecuzione. Il secondo argomento del comando LN è la libreria standard fornita con il linguaggio, necessaria dal momento che abbiamo usato la funzione "printf" all'interno del nostro sorgente.

Per l'esecuzione è sufficiente scrivere "TEST" sul terminale e dare l'invio.

Per generare la versione pseudocode del programma è necessario usare il compilatore CCI al posto di C65:

```
CCI TEST.C
```

Questi produce, dopo la chiamata all'assemblatore, un codice rilocabile test.int linkabile con le librerie:

```
LN TEST.INT SAINT.LIB
```

Le librerie da usare, in questo semplice esempio solo la SAINT.LIB, sono le equivalenti di quelle "65" da usare per il linking "nativo".

Nel nostro particolare esempio il codice ottenibile con il compilatore CCI è più grande rispetto a quello di C65. Si tratta di un effetto collaterale al fatto che l'eseguibile in se e per se è la minima parte rispetto al codice delle librerie linkate.

Non abbiamo notato differenze nell'esecuzione nei due casi.

Conclusione

L'Atzec C, come ambiente di produttività, meriterebbe ben altra trattazione. In generale questo primo approccio e l'uso di qualche giorno da parte nostra, ci permette di promuovere a pieni voti questo prodotto della MANX Software. Con esso il programmatore Apple dispone di un discretamente completo ambiente di sviluppo in grado di generare codice nativo, pseudo-compilato o anche misto, in grado di coniugare le esigenze di efficenza e affidabilità che si richiedono ad un ambiente di sviluppo.



L'approccio è facilitato dalla conoscenza dell'ambiente UNIX e dei suoi comandi (pensiamo all'editor ad esempio) facilita enormemente l'apprendimento da parte del programmatore provvisto dello skill Unix, mentre il classico coder Applesoft dovrà mettere in conto una certa quantità di giornate per imparare linguaggio e shell.

La relativa povertà dei comandi della shell e alcune scelte non proprio logiche (la gestione dei drive, almeno nella versione DOS 3.3), possono risultare limitanti se non fastidiose, ma non va dimenticato che siamo su una classe di sistemi molto lontana dalle workstation UNIX e obiettivamente sarebbe stato difficile fare di meglio.

Bibliografia

Atzec C Ufficial Museum:

http://www.clipshop.ca/Aztec/

Atzec C user's manual.

[Sm]

Un sito ben curato e completo per chi vuole approfondire e trovare software e documentazione sull'Atzec C Compiler.

```
APPLE ][ SHELL 2.4

COPYRIGHT (C) 1983
BY MANX SOFTWARE SYSTEMS
```

Emulazione

SMILL

I mondi virtuali a volte possono essere molto realistici...

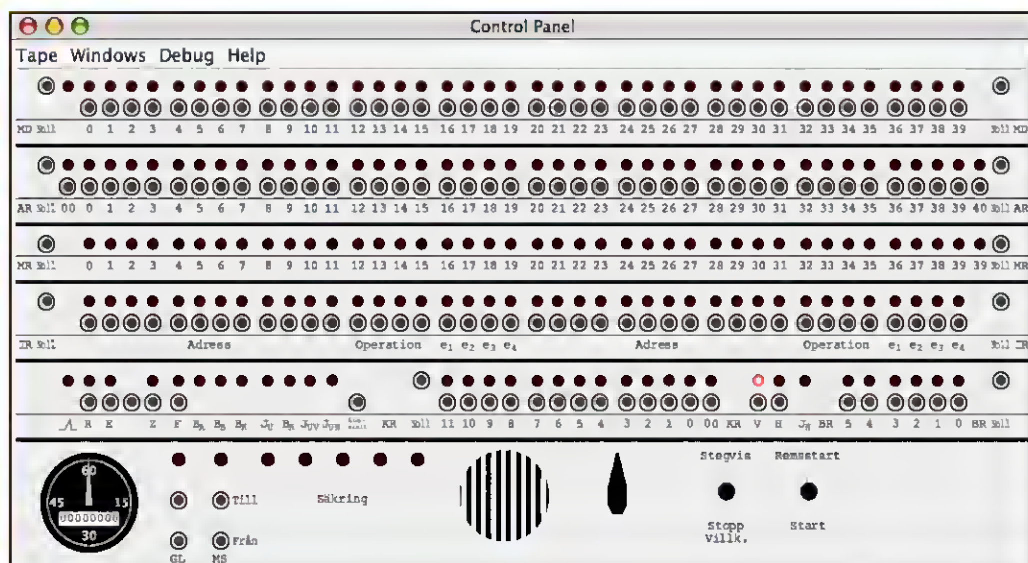


Figura 1.
Il pannello di controllo del sistema. Lampadine e pulsanti al posto di schermo e tastiera...

Introduzione

Sì lo so cosa state pensando: -"Cos'è 'sta roba?". E' un emulatore, ovvio. Sì, ma di che? Di un computer, ovvio al quadrato!

Dunque vediamo di andarci con calma. Sicuramente saprete che anche i computer hanno le loro generazioni, ora siamo alla quarta generazione (o alla quinta?), comunque prima di questa c'è stata la prima generazione, la seconda generazione e così via.

Io SMILL è un computer di prima generazione. Semplice no?

Dire che un calcolatore è di prima generazione è facile: sono i primi costruiti! Più difficile è decidere quando finisce la prima e comincia la seconda: tutte le fasi di transazione sono problematiche, basta pensare all'adolescenza! Scusate l'esempio: sono alle prese con l'adolescenza di mia figlia e non è affatto facile... Chi ci è passato mi capisce.

Come era fatto un computer di prima generazione e come funzionava? Credo che tutti gli appassionati di informatica abbiano visto le foto dei vari ENIAC e ne siano rimasti impressionati per le dimensioni, oltre che irriderne le prestazioni. L'ENIAC e compagnia non era il solo calcolatore esistente. Più o meno in tutto il mondo della ricerca si lavorava all'idea del calcolo automatico. In Svezia una equipe di scienziati all'Università di Lund nel 1956 si mette a costruire una macchina programmabile: lo SMILL, che starebbe per "Sif-

fermaskinen i Lund" (così si scrive, come si pronunci mi è sconosciuto). La traduzione del nome sarebbe pressappoco: "Macchina digitale di Lund".

SMILL è un miglioramento del precedente e primo progetto assoluto di un calcolatore in Svezia. Tale macchina era chiamata BESK-1 (Binär Elektronisk Sekvens-Kalkylator) ed è del 1953. Siamo veramente ai primissimi esperimenti!

Per certi versi possiamo dire che gli studi sulla computazione automatica nel primo dopoguerra avevano un filone promettente nei paesi del nord Europa. Analoghi esperimenti di quelli svedesi si trovano in Norvegia, tutti che prendono spunto da un documento di ricerca che ne delineava i principi fondamentali e l'architettura che dovevano seguire queste macchine. Questo documento si trova on-line, per chi volesse approfondire l'argomento, all'indirizzo http://bitsavers.trailing-edge.com/pdf/ias/IAS_Final_Report_Jan54.pdf; il titolo del documento è "Final Progress Report on Phisical Realization of an Electronic Computing Instrument". Il coordinatore del progetto era un certo "John Von Neumann" (vi ricorda qualcosa, vero?).

Il calcolatore SMILL

L'architettura del sistema rispecchia le idee del suo ideatore teorico: dati e istruzioni sono memorizzati nella stessa memoria ed esiste una "CPU" con registri e unità di calcolo.



Figura 2.
Il predecessore dello SMILL, chiamato BESK-1. Mancono foto della macchina SMILL, ma più o meno dovrebbe essere stata simile.

Inizialmente il calcolatore era molto primitivo: 2048 locazioni di memoria su cilindro (magnetic drum), input via nastro di carta (punch reader) e output su stampante. Nella seconda versione la memoria viene raddoppiata e le 4096 locazioni coprono tutto lo spazio di indirizzamento della macchina (che è di 12 bit); il cilindro magntico è sostituito con memoria "core" (probablmente a nuclei di ferrite, ma la documentazione non lo specifica), rimangono lettore a nastro e una stampante dedicata all'output, ma compare un pannello di controllo (quello che vedete nell'immagine di apertura) che permette l'inserimento delle istruzioni piuttosto che limitarsi a leggerle da nastro.

Le locazioni di memoria sono a 40 bit, numerati all'opposto della convenzione attuale, cioè il bit 0 è il più significativo e il bit 39 è quello meno significativo. le istruzioni sono invece a 20 bit e quindi ogni locazione di memoria ne contiene due. I registri sono a 40 bit mentre

il program counter ne ha 13: 12 per l'indirizzo di memoria e un bit per discriminare fra l'istruzione immagazzinata nella parte alta o in quella bassa.

I registri sono in totale 5. Oltre al già citato program counter, individuato dalla sigla KR, esiste un registro IR che contiene la coppia di istruzioni prelevate dalla memoria, il classico accumulatore AR e due registri di calcolo chiamati MD e MR, rispettivamente per "moltiplicando" e "moltiplicatore".

Bisogna pensare che la macchina è costruita solo per fare calcoli, non per trattare informazioni più generali come ora concepiamo i PC. Così si spiega la presenza di due registri esplicitamente dedicati alle istruzioni di calcolo.

Come SMILL gestisce la rappresentazione numerica non è un tema che possiamo affrontare in questo contesto, diciamo brevemente che si rappresentano i numeri da -1 a +1 a virgola fissa con una precisione di 29 cifre.

L'emulatore

L'emulatore è stato costruito per celebrare il 50^ anniversario della costruzione dello SMILL all'università Lund, con il preciso intento di rendere alla perfezione l'atmosfera tecnica che si respirava all'epoca.

Esiste in due versioni: Java ed eseguibile per MAC OSX. Abbiamo scelto quest'ultima versione che ci sembra più curata nell'aspetto grafico. La versione è la 1.1, prelevabile all'indirizzo dedicato al progetto (www.smillemu.org).

L'applicazione dispone di più finestre che lavorano assieme: una è il pannello di controllo principale, un'altra mostra un punch reader virtuale (figura 5) che può essere caricato con alcuni tape predefiniti (figura 4). nella figura 3 si hanno

Figura 3.
La finestra che mostra il contenuto dei registri della macchina.

Registers

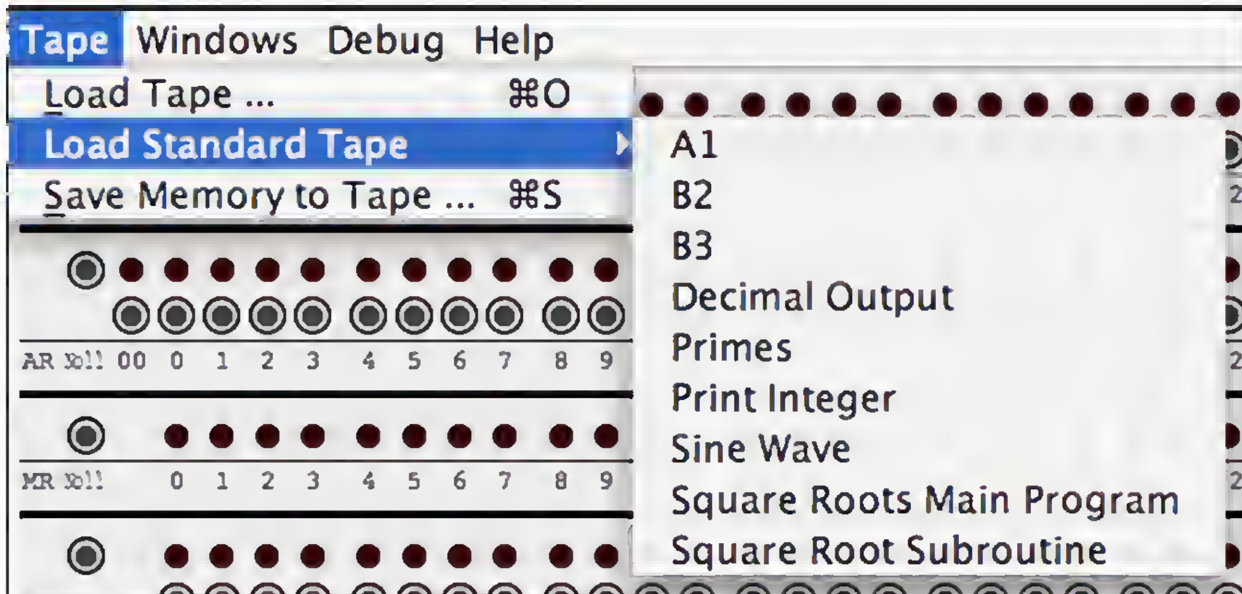
MD
02AAA AAAAA = 0.0208333333321207

AR
1 01460 EA912 0 = 0.0099504781283031

MR
00000 00000 = 0.0000000000000000

IR
Left 40CA8 JUMP 40c left if AR < 0
Right 000F8 PRINT '<space>'

KR
40A 0



a disposizione i registri in una comoda visualizzazione "moderna", tutt'altra cosa che decifrarne il contenuto dal pannello di controllo!

Uso

Qui non c'è una tastiera, si lavora direttamente sul pannello frontale della macchina (quello che abbiamo riprodotto nella schermata di inizio articolo).

Il pannello di comando è formato da luci (i bollini scuri pieni) e pulsanti (i bollini grigi cerchiati). Le luci si accendono per segnalare la condizione 1 del bit corrispondente e si spengono ad indicare lo zero logico. Quando si preme un bottone la luce si accende, per spegnerla è necessario resettare l'intera riga con uno dei pulsanti predisposti uno a sinistra della riga e l'altro a destra.

Come vedete il pannello è diviso orizzontalmente in sei parti, ben distinguibili dalla linea in grassetto che separa i singoli pannelli.

Il primo pannello in alto rappresenta il registro MD (moltiplicatore) della macchina. I suoi bit, numerati da 0 a 39, partendo da sinistra, possono essere impostati operando con i rispettivi pulsanti.

Il pannello sotto è l'accumulatore AR, quello successivo è il moltiplicando che non si imposta a mano ma con l'uso delle istruzioni.

IR è il registro delle istruzioni: due per ogni fase di fetch. Il successivo pannello è piuttosto denso di indicatori, compresi i bit del program counter KR (a 12 bit, come si diceva).

Infine il pannello più basso con una specie di orologio (è un contatore di tempo di funzionamento), la feritoia dell'altoparlante e altri pulsanti compresi due deviatori siglati

Figura 4.
Alcune "tape" predefinite, ma altre possono essere salvate e caricate.

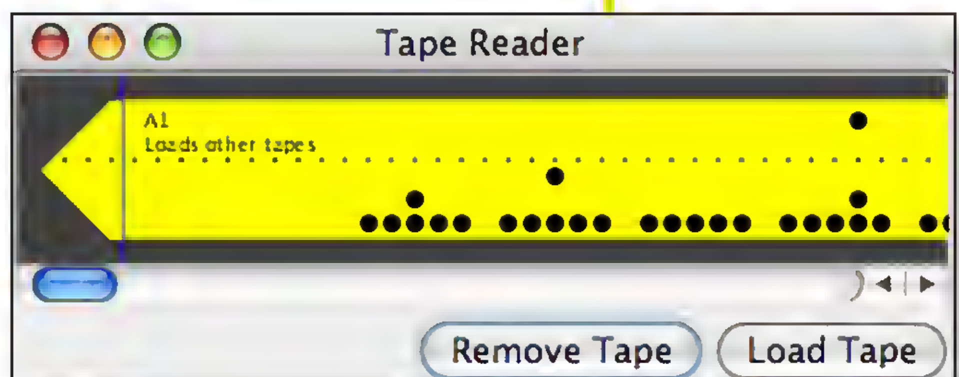


Figura 5.
La curiosa periferica a nastro di carta chiamata "punch reader". Codifica 4 bit alla volta per ogni posizione di avanzamento del nastro.

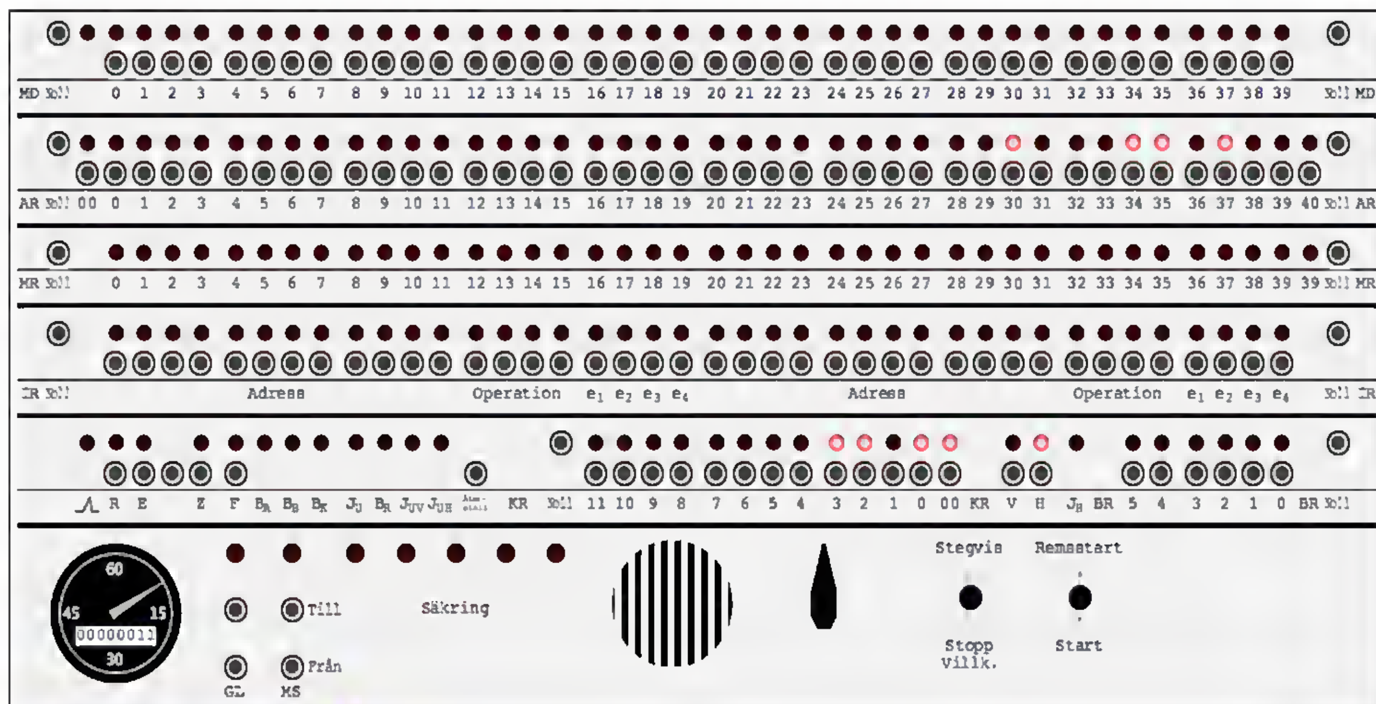


Figura 6.
Dopo il caricamento del nastro di "bootstrapp". Quello che conta è l'indirizzo del program counter (registro KR), pronto per eseguire il caricamento di un'altro nastro.

"stopp" e "start".

La fila di luci sopra la scritta 'Säkring' indica il funzionamento corretto di ognuna delle cinque tensioni di alimentazione.

All'estrema destra di questo pannello si trova un pulsante a due posizioni con le sigle 'Remsstart' e 'Start'. Nella posizione in alto effettua la sequenza di bootstrapp, in posizione down (su 'Start') esegue il programma a partire dall'indirizzo di memoria contenuto nel program counter e fino a che non si incontra una posizione di stop.

Il pulsante di reset è quello siglato 'Återställ' (è svedese, scusate molto :-).

E' interessante notare come nemmeno l'autore dell'emulatore, che pure deve avere studiato ogni documentazione disponibile, è in grado di dire a cosa servissero tutti i pulsanti! Ecco che qualcosa si è perso e, a meno di non trovare altra documentazione o qualche vecchietto che ci abbia lavorato all'epoca (ma ora dovrebbe avere almeno ottanta anni), sarà difficile recuperare una conoscenza piena di questa macchina. L'autore suppone che qualche pulsante e di conseguenza qualche funzione associata, derivino da un progetto forse diverso, magari realizzato solo in parte. Non dimentichiamoci che la componente meccanica del sistema è piuttosto importante e quindi una volta messo un pulsante e una lampadina non è che si possa buttare via il pannello per rifarlo: sarebbe inutilmente costoso per un sistema destinato alla ricerca.

Tornando al funzionamento del nostro SMILL, l'idea è quella che prima si carica il programma in memoria, compresi gli eventuali dati, poi si procede con l'esecuzione ottenendo il risultato sulla stampante o anche solo semplicemente come valore nei registri. Quando parliamo di programma intendiamo ovviamente una sequenza di

istruzioni binarie. Non c'è spazio e tecnologia sufficiente per usare un compilatore, nemmeno un semplice assemblatore.

L'alternativa è quella di affidarsi al nastro: si carica in memoria e si esegue il codice. Bravo, direte voi: -"le istruzioni per caricare il nastro saranno come minimo qualche decina!". Vero, qui però i progettisti lo sapevano ed ecco che hanno programmato un nastro che parte al l'accensione della macchina e carica un programma in grado di leggere gli altri nastri; un bootstrapp vero e proprio, insomma!

Quindi, predisposto il nastro A1 (si trova nella lista dei nastri standard), si posizione il deviatore su "Start" per caricare il programma dal nastro alla memoria.

L'emulatore fa scorrere il nastro durante la lettura e una linea blu verticale indica la posizione della testina di lettura del lettore: è una animazione divertente da vedere che dimostra la cura e il piacere provato dall'autore nel realizzarlo.

Ora lo Smill è pronto a caricare un'altro nastro (figura 6).

In figura 7 la finestra di dump della memoria. Per ogni indirizzo sono mostrate le due word in esadecimale (ricordate che ogni locazione contiene due parole?). nella colonna "Values" due righe: la prima con il codice esadecimale delle due parole e la seconda con l'interpretazione decimale dei 40 bit. Infine nella colonna "Instruction" il mnemonico dell'istruzione (destra e sinistra).

Ad esempio all'indirizzo 0 (figura 7) troviamo 40 bit che significano due istruzioni:

Tape word -> AR

JUMP 007 left

oppure il numero decimale 0,000488...

Essendo lo Smill una macchina con architettura Von Neumann, è il contesto che discrimina il significato dei byte in memoria.

Ultima osservazione sulla istruzione JUMP 007 left. Perché "left"? perché all'indirizzo 007 ci sono due parole e la prossima da eseguire dopo il jump è in questo caso quella di sinistra.

Decidiamo di provare il programma "Primes" che stampa i primi 128

Figura 7.
Il dump della memoria; una comoda funzione per esaminare le istruzioni in esecuzzione.

Address	Value		Instruction
000	00100 00790		tape word -> AR, [001]
	0,0004882847715635		JUMP 007 left
001	00000 00234		tape word -> AR, [000]
	0,0000000010259100		AR{0..11} -> [002]{0..11}
002	00000 00036		tape word -> AR, [000]
	0,0000000000982254	0 -> AR,	0 -> [000]{0..11}
003	00852 00250	0 -> AR,	[008] -> AR
	0,0040626536647324		AR + [002] -> AR
004	00230 00858		AR -> [002]
	0,0010681191197364		AR - [008] -> AR
005	014DC 00058		AR >> 20 -> AR
	0,0101852418592898		AR - [000] -> AR
006	002A8 000B8		JUMP 002 left if AR < 0
	0,0012969974050066		NOOP
007	000B0 00192		HALT
	0,0003356940906087	0 -> AR,	JUMP 001 left
008	00100 00000		tape word -> AR, [001]
	0,0004882812500000		tape word -> AR, [000]
009	00000 0000A		tape word -> AR, [000]
	0,0000000000181899	0 -> AR,	tape row -> AR{36..39}
00A	40000 00000		tape word -> AR, [400]

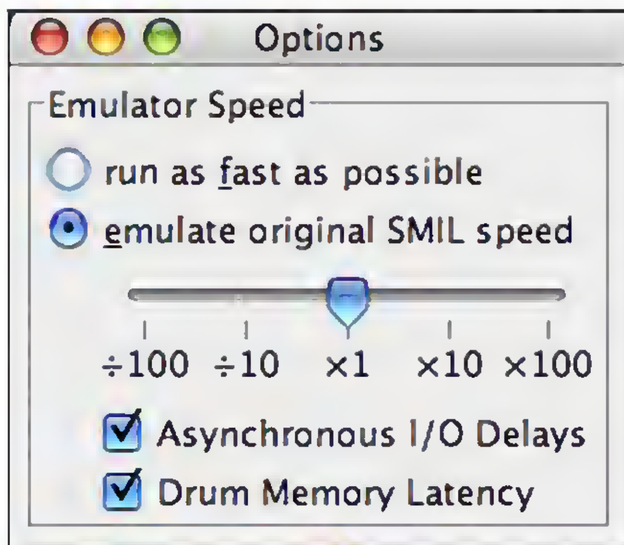


Figura 8.
Si può impostare la velocità di esecuzione. Quella originale è la migliore per avere un autentico return to back.

numeri primi.

Serve anche la routine di stampa di un numero in decimale. Quindi carichiamo prima questo nastro e poi quello di calcolo della sequenza. Ogni nastro può essere caricato in sequenza operando con il bottone "Start", dato che ogni caricamento precedente lascia comunque la memoria predisposta per il caricamento dei nastri successivi.

Questa sequenza di caricamento di codice va a riempire le locazioni di memoria occupate da ciascuna routine. Alla fine mettiamo il nastro "Primes" nel lettore e carichiamo.

Ora che abbiamo finito di preparare la memoria lanciamo il vero programma. Resettiamo il program counter (registro IR) e il registro KR. Predisponiamo l'indirizzo di partenza (esadecimale 20, binario 0100 0000 0101 1001 0000) in IR, accendendo le corrispondenti luci.

Infine... Start e vai!!!

La velocità di esecuzione è impostabile da menù, ma ovviamente il migliore feedback si ottiene lasciando la velocità originale (vedi figura 8).

Durante l'esecuzione il sistema lampeggia, in sintonia con il valo-

re che ogni registro assume nelle varie istruzioni. Il pannello lampeggiante assomiglia proprio a quelle immagini stereotipate dei cervelli elettronici di certa filmografia anni settanta. Ecco da dove veniva l'idea: dallo Smill!.

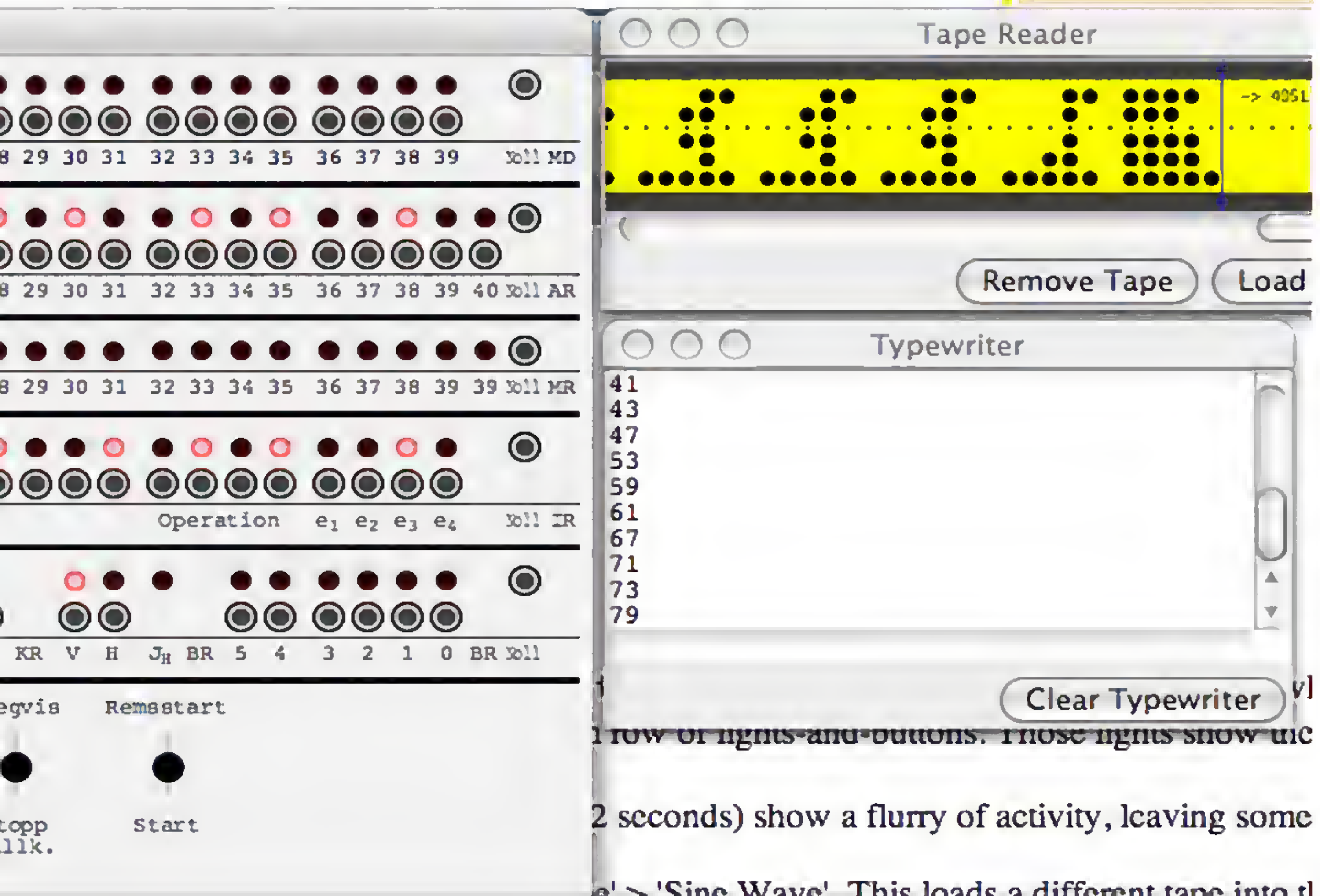
Conclusioni

Divertente. Ecco l'aggettivo giusto per descrivere questo emulatore. Ovvio che dovete essere "topi da retro-garage" per ricavarne vero godimento!

Otre a questo piacere personale rimane un valore molto elevato di questo progetto che è appunto quello di tramandare una conoscenza importantissima. Il fatto che l'autore si dichiari non in grado di ricostruire al 100% il funzionamento, la dice lunga sulla volatiilità delle conoscenze tecniche legate allo sviluppo dell'elettronica digitale. le cose che sono ovvie oggi potrebbero non esserlo domani, fra meno di un secolo...

[L2]

Figura 9.
Una immagine wide del sistema durante il caclolo dei numeri primi. Il pannello che lampeggia durante l'esecuzione e l'utout sulla stampante mano a mano che i numeri primi sono trovati dal programma.



L'intervista

Vari personaggi e amici incontrati qua e là per una chiacchierata sul mondo del retro computing.

Intervista ad Andrea

Abbiamo incontrato Andrea in occasione di una mostra-mercato dedicata al consumo equo e solidale. L'iniziativa si chiama "Fà la cosa giusta" e mi sembra proprio che gli organizzatori abbiano colto nel segno scegliendo questo slogan diretto che fà riflettere (o almeno dovrebbe) sui nostri comportamenti quotidiani più banali, come l'andare a fare la spesa.

Andrea, assieme con altri amici, aveva organizzato un banco dimostrativo della sua associazione che si occupa di trashing e l'occasione è stata buona per rivolgergli alcune domande.

[JN]

Prima di tutto potresti spiegare in due parole cos'è il trashing?

[Andrea]

Sono tutte quelle attività rivolte al riuso delle tecnologie dismesse, in particolare nell'informatica. Sappiamo come invecchiano presto i PC e la maggior parte di essi finisce in discarica dopo tre/quattro anni. Noi cerchiamo di recuperarli e riutilizzarli.

[JN]

Quindi frugate nei cassonetti e cercate di ricavare qualcosa da quello che altri hanno buttato. Ma conviene?

[Andrea]

Questo è il punto chiave. Conviene dal punto di vista economico ma soprattutto dal punto di vista ambientale. Economico perché tendiamo a dimostrare che si può utilizzare un PC ben oltre la sua vita considerata "normale" e ambientale, perché meno prodotti vanno in discarica e più ci guadagna l'ambiente.

[JN]

Ma se uno ha deciso di cambiare il PC, non è perché è obsoleto rispetto alle sue necessità? Voi lo recuperate per darlo ad associazioni che non possono permettersi un PC di ultima generazione?

[Andrea]

Sì e no. Inizialmente l'idea del trashing era quella di consentire ai meno fortunati un utilizzo tecnologico appena inferiore al top ma comunque valido. Ora è diventato quasi predominante il tentativo che facciamo nel dimostrare che non è necessario cambiare il PC. Se da un certo punto di vista prima del 2000 noi stessi incoraggiavamo il cambio dei PC per avere materiale migliore da riciclare, ora il punto di vista è cambiato e il nostro inte-

resse primario è evitare che si producano rifiuti inutili e dannosi per l'ambiente.

[JN]

Quindi si può affermare che il trashing è una pratica ecologica...

[Andrea]

E' così! Il nostro goal è evitare che siano acquistate tecnologie invasive semplicemente perchè il mercato ne stimola la vendita.

[JN]

E le persone si convincono della vostra idea?

[Andrea]

Queste iniziative di divulgazione hanno proprio lo scopo di far conoscere un punto di vista diverso e stimolare il consumo critico della gente. La maggior parte degli utilizzatori non conosce le tecniche più elementari per allungare la vita attiva di un computer. La lamentela più frequente è che il PC è diventato lento e che per questo va cambiato. Un'altra è che un certo gioco non funziona perché la scheda grafica è obsoleta. Qui hanno buon gioco i venditori ovviamente, il cui interesse è vendere una macchina intera e non asistere il cliente ad esempio ad una riconfigurazione o svecchiamento hardware della sua macchina.

[JN]

E voi riuscite a convincere qualcuno?

[Andrea]

facciamo queste iniziative proprio per questo. Abbiamo fatto anche delle giornate che abbiamo chiamato "renew-day" durante le quali chiunque può portarci l'unità centrale o il suo portatile e i nostri soci reinstallano l'intero ambiente operativo, applicativi compresi e il "cliente" se ne va con una macchina che sembra nuova, almeno per la velocità di esecuzione.

Non è raro che semplicemente ci si trovi con un PC talmente infestato da malware da renderlo inutilizzabile: basta una scansione antivirus e tutto riprendde a funzionare...

Il problema è che queste attività occupano un sacco di tempo e l'utente spesso viene con il computer ma non ha idea se possiede o meno la licenza per Windows o per l'Office...

[[JN]

E voi come vi comportate in questo caso?

[Andrea]

Non possiamo installare software non originale, ovviamente! Riusciamo a volte a convincere qualcuno ad usare Linux e Open Office piuttosto che il software proprietario, ma è una battaglia difficile perché si debbono superare degli scogli culturali difficili da abbattere.

[JN]

Quindi mi dici che per risparmiare bisogna che l'utente sia disposto a rinunciare a qualcosa, fosse solo la sua idea di utilizzo del PC come l'ha usato da sempre.

[Andrea]

Questo è vero relativamente. Da una parte è vero che l'utente tende ad usare Windows perché l'ha sempre avuto sulla macchina, ma quando vede il suo vecchio PC "volare" con Linux, si convince che esistono prodotti che possono essere migliori di quelli tanto pubblicizzati.

I successi migliori li abbiamo con le nostre proposte di aggiornamento nelle quali convinciamo una persona che non serve cambiare l'intera macchina e che basta comprare una scheda video più aggiornata per farci girare quel gioco che gli interessa tanto.

[JN]

Già, i giochi... se non ci fossero probabilmente i PC durerebbero ben oltre i tre anni attuali...

[Andrea]

Questo è vero, anzi si può tranquillamente affermare che i PC venduti sarebbero la metà

se non ci fosse il comparto game. E' ben vero che la gente tende oggi a comprare le console, se ha interesse solo per il gioco, ma il PC è ancora la piattaforma più diffusa per il gaming.

[JN]

Appunto sulle console, fate recupero anche di queste?

[Andrea]

Ci limitiamo a raccogliere quelle che ci donano con i relativi titoli, le ricondizioniamo (per così dire) e le diamo a qualche organizzazione che si occupa ad esempio di distribuire giochi ai bambini meno fortunati. Non è un comparto nel quale ci piace operare: è abbastanza chiuso e gli interventi che si potrebbero mettere in atto per "svecchiare" il sistema sono tutti illegali purtroppo.

[JN]

Quindi se ho ben capito i sistemi sui quali puntate sono i PC di penultima generazione.

[Andrea]

Sono le macchine che ci danno maggiore soddisfazione, ma non disdeniamo il riutilizzo di hardware anche più vecchio. Purtroppo le macchine fino a 1 MHz, tanto per dare un confine, possono incidere poco nell'utilizzo "domestico", se non limitatamente a situazioni "di nicchia", ad esempio ci vengono chiesti spesso dei portatili minimi o per la scrittura della tesi di laurea o altri lavori molto limitati e per i quali serve ad esempio un solo software o un collegamento alla rete tipo "terminale".

Le tecnologie terminal, che avrebbero potuto essere una via di sfogo per il riutilizzo dell'hardware obsoleto, non hanno conosciuto finora la diffusione che si preannunciava quanche anno fà.

[JN]

Secondo te per quale motivo?

[Andrea]

Probabilmente perché sono costose alla fin fine: servono fior di server per supportare i desktop virtuali degli utenti. In ogni caso il costo di un PC minimo da usare come terminale è talmente basso che non conviene nemmeno ripiegare su un ingombrante minitower, magari rumoroso e mangiatore di elettricità, al posto di uno dei nuovi scatolotti da poche centinaia di Euro...

[JN]

E che mi dici delle altre tecnologie? Mac, i laptop, i palmari,...

[Andrea]

Si potrebbe fare molto, ma servirebbero ben altre forze e competenze. Noi ci limitiamo ai PC proprio perché non avremmo la forza di occuparci d'altro. Prendiamo ad esempio le stampanti: sai quante ne vengono buttate? Addirittura più degli stessi computer! E' ovvio: costano pochissimo, hanno la sfiga di avere una componente meccanica predominante, e quindi si rompono più facilmente, e la qualità di stampa che offrono è in continua crescita. Difficile fare qualcosa, se non recuperare le migliori e darle in dotazione ai PC che approntiamo per associazioni o enti senza scopo di lucro.

Abbiamo un buon successo con gli scanner invece. Paradossalmente lo scanner viene cambiato non per le prestazioni ma perché diventato obsoleto dal punto di vista della tecnologia di comunicazione (ad esempio perché richiede una interfaccia SCSI o magari una parallela, sparita nei moderni PC). Oppure, cosa che abbiamo scoperto di recente, perché il costruttore non rilascia i driver per i vecchi modelli. In questo caso mettiamo in campo le nostre conoscenze e non è difficile far "digerire" uno scanner di qualche anno fa nemmeno al nuovo Vista...

[JN]

E le altre componenti: dischi, floppy, etc...?

[Andrea]

Per tutto teniamo una buona scorta, ma certo che aldilè del rimpiazzo uno-a-uno di un

componente guasto, non è che un vecchia HD ide possa giocare granché sul terreno del riuso...

Abbiamo recentemente incontrato un discreto interesse per le chiavette USB, le pen-drive per capirci. La gente le cambia spesso perché ha bisogno di una maggiore capacità: si sà che l'appetito vien mangiando.

Però una pen drive da un giga è ancora validissima ad esempio per uno studente delle elementari o delle medie... Certo che questo materiale costa talmente poco che non viene in mente quasi a nessuno di chiedere in giro se c'è dell'usato.

[JN]

Mi sembra di capire che il sistema operativo GNU/Linux abbia una parte non marginale nella vostra azione.

[Andrea]

Direi che è inevitabile che sia così: del resto ha un vantaggio non indifferente sui vari WIndows: è gratis e quindi ci permette di installarlo su qualsiasi cosa senza incappare nelle violazioni delle licenze. Facendo questa attività si scoprono cosucce mica male sulle licenze software, ad esempio che tu puoi benissimo regalare un PC ad un amico o alla nostra associazione, ma non puoi regalargli il sistema operativo e i programmi che sono installati!

Per non correre rischi noi preferiamo cancellare tutto e installare solo se abbiamo la licenza originale, oppure diamo il PC ricondizionato ma pulito-pulito e il nostro utente poi si deve arrangiare o assumersi in prima persona eventuali responsabilità.

L'unica cosa che possiamo fare, come dicevo, è incoraggiare al massimo le tecnologie open, ma non sempre è facile...

[JN]

Adesso voglio fare la parte dell'avvocato del diavolo e ti chiedo: ma ha senso mettersi in casa un PC vecchio che comunque è più lento, consuma di più e è infinitamente più rumoroso?

[Andrea]

Perché faresti una cosa buona per l'ambiente, ti basta? Sul consumo e la rumorosità posso darti ragione ma si può intervenire ad esempio cambiando le ventole con una serie meno rumorosa e per il consumo cercando di usare una macchina adeguata al proprio lavoro e non sovradimensionata per clock, dischi o memoria che consumano inutile energia.

Non abbiamo trattato il tema degli schermi CRT, in effetti. Questo è un comparto difficile da rivalutare: diciamo che si tratta di una tecnologia veramente obsoleta e che è bene ci si sia liberati: i nuovi schermi LCD consumano meno e alla fin fine durano di più del tubo catodico oltre ad avere altri vantaggi obiettivi: meno consumo e più spazio sulla scrivania. Ecco, questa forse è una tecnologia da non rimpiangere, anche se ci sono schermi CRT bellissimi e la cui qualità solo ora è avvicinata dalle tecnologie LCD.

[JN]

Grazie per la chiacchierata, ho imparato un sacco di cose...

[Andrea]

Questo è il nostro scopo primario: comunicare.

[JN]

Mi chiedo se il trashing avrà un futuro con la costante diminuzione dei costi delle tecnologie...

[Andrea]

Credo ci sarà sempre spazio per la nostra attività, le nostre idee e le nostre iniziative. L'informatica personale ha ancora zone "grige" che noi vorremmo illuminare. Mi riferisco al fatto che il marketing impone un consumo a volte non giustificato e per il quale poi paghiamo le conseguenze tutti in termini di ambiente e in definitiva di qualità della nostra vita.

Retro Linguaggi

La storia dell'informatica è stata anche la storia dei linguaggi di programmazione.

ABAP (parte 10)

In questa lezione del corso ABAP impareremo alcune cosette che riguardano l'interfaccia grafica. Quello che abbiamo realizzato la volta scorsa era effettivamente molto elementare: tre campi e due bottoni.

Chiaro che una Gui che si rispetti dovrà attivare dei menu, dei bottoni su una toolbar, e attivare dei tasti funzione... insomma tutti gli elementi che siamo ormai abituati ad utilizzare in un qualsiasi programma Windows, ed è quello che faremo in questa lezione.

Per incominciare riprendiamo il sorgente del nostro programma cominciato la scorsa volta ed aggiungiamo una doverosa segnalazione. Cioè mettiamo sull'interfaccia un campo "Messaggio" che ci avvisi se l'operazione di inserimento del record anagrafico è andato a buon fine oppure no.

Nella figura 1 la definizione del campo all'interno dello Screen Painter. Da notare che abbiamo impostato "Output only" nelle proprietà del campo stesso.

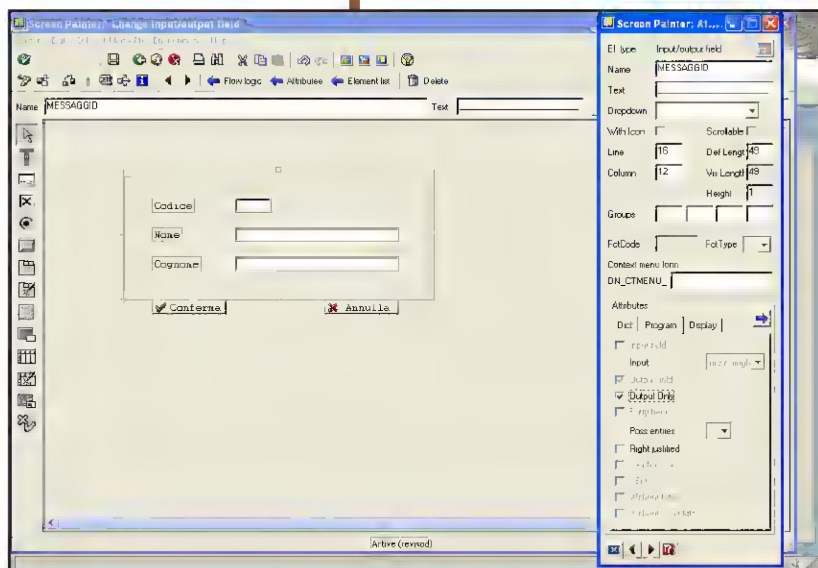
Abbiamo bisogno di una variabile di nome "MESSAGGIO" all'interno del programma e dobbiamo impostare il suo valore in relazione al risultato dell'operazione comandata.

Il listato 1 riporta il codice del programma con le modifiche che riguardano la gestione appena detta. In particolare c'è il controllo dell'avvenuta operazione sul DB e conseguente impostazione del messaggio.

Quindi siamo arrivati ad una dimostrazione importante: quando i campi nella maschera video hanno lo stesso nome delle variabili globali nel programma, allora c'è una diretta corrispondenza di quanto vediamo a video e di quanto registrato nel valore della variabile corrispondente.

A questo punto parliamo degli altri elementi di interfaccia: menù, toolbar, etc... che ci servono per com-

Figura 1.



pletare l'interazione con l'utente.

Tutti questi elementi sono organizzati in una struttura che viene chiamata "GUI Status" e che viene assegnata allo screen prima della sua emissione a video.

Attiviamo quindi la gestione del nostro screen 100 e abilitiamo il modulo di gestione dell'evento Process Before Output (Figura 2).

Con un doppio click sul nome del modulo (la proposta è STATUS_100), creiamo le corrispondenti righe nel codice del programma principale.

```
MODULE STATUS_0100 OUTPUT.
  SET PF-STATUS 'STATUS_100'.
*   SET TITLEBAR 'xxx'.

ENDMODULE.
```

Ci sono due istruzioni predisposte, decommentiamo la prima scegliendo un mome per lo status (la nostra scelta è stata "Status_100", ma qualunque nome va bene).

Ora lo status deve essere creato e per questo esiste un tool apposito che si raggiunge con un doppio click sul nome stesso.

Uno status è formato da tre elementi principali: il menù dell'applicazione, le toolbar (quella di sistema subito sotto il menù e quella dell'applicazione subito sopra la client area della finestra utente ed infine il terzo elemento che sono le funzioni associate ai vari tasti, compreso ENTER, ESC, etc... ma soprattutto le varie combinazioni dei tasti F1...Fxx sulla tastiera.

Questi tre elementi principali si possono "aprire" per vederne i dettagli. Nella figura 2 abbiamo aperto la definizione del menù e cominciato ad aggiungere le voci corrispondenti: la voce "File" dove abbiamo predisposto la scelta per uscire dal programma e il menù "Edit" con la funzione di conferma, equivalente al bottone "Conferma" sull'interfaccia.

```
PROGRAM ZMX_09_001                    .

TABLES: ZMX_TAB_001.

DATA FCODE TYPE SY-UCOMM.
DATA MESSAGGIO(40) TYPE C.


MODULE USER_COMMAND_0100 INPUT.

 DATA  RCODE TYPE SY-UCOMM.

 MOVE FCODE TO RCODE.
 CLEAR RCODE.
 CLEAR MESSAGGIO.

 CASE FCODE.
  WHEN 'CONF'.
   INSERT ZMX_TAB_001.
   IF SY-SUBRC > 0.
    MOVE 'Errore nell''inserimento del record'
     TO MESSAGGIO.
   ELSE.
    MOVE 'Operazione eseguita correttamente'
     TO MESSAGGIO.
   ENDIF.

  WHEN 'ANNULLA'.
   LEAVE PROGRAM.

 ENDCASE.

ENDMODULE.
```

Listato 1.

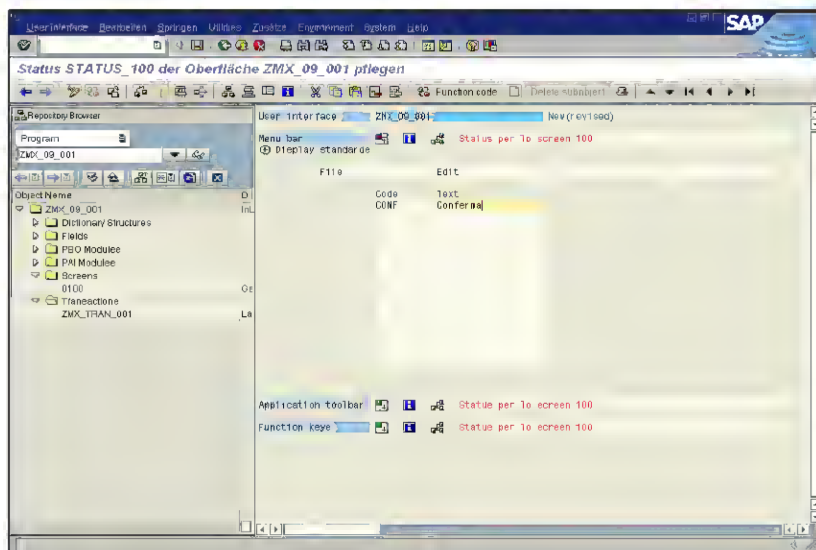


Figura 2.

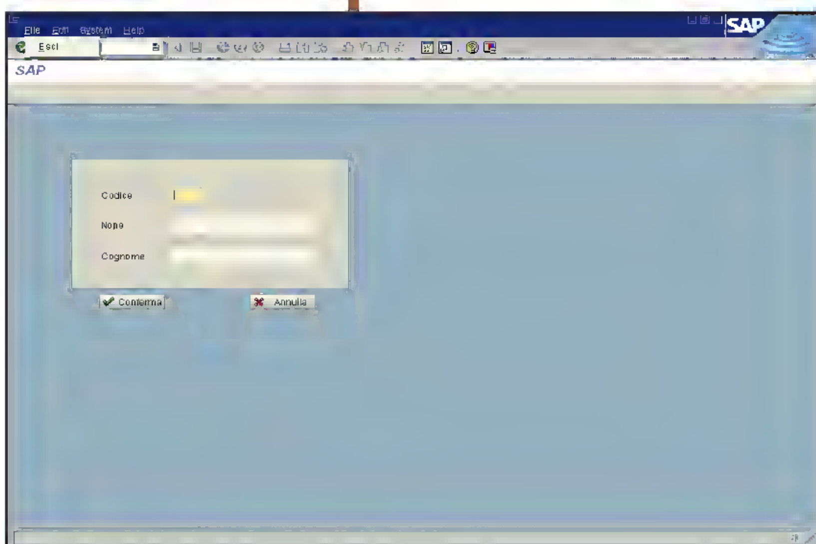
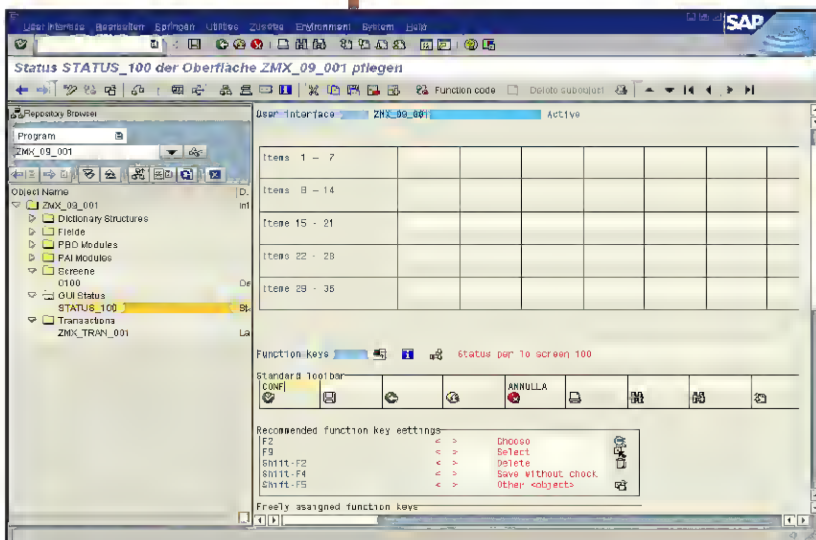


Figura 3.

Figura 4.



La definizione dei menù è elementare: sotto la colonna "Code" si inserisce il valore del codice che deve essere ritornato al programma; sotto la colonna "Text" trovano posto le descrizioni che appariranno all'apertura del menù.

Anche lo Status deve essere attivato per essere utilizzabile dal programma.

Nella figura 3 l'effetto di abilitazione del menù con le due funzioni "File" e "Edit" aggiunte dallo status definito.

Ora attiviamo i bottoni sulla toolbar di sistema, come ad esempio il bottone di uscita (quello rosso con la "X") e quello di conferma.

Per fare questo ri-editiamo lo status, inserendo nella "Standard Toolbar" i valori delle funzioni in corrispondenza delle icone predisposte (figura 4).

Nella figura successiva abbiamo abilitato anche la toolbar dell'applicazione che però necessita di associare un tasto funzione alla funzione da abilitare stessa (noi ci abbiamo associato il tasto F2).

Infine nella pagina a fronte la figura 6 mostra il complesso delle modifiche apportate al programma con le funzioni di status abiltate secondo quanto abbiamo predisposto.

[Mx]

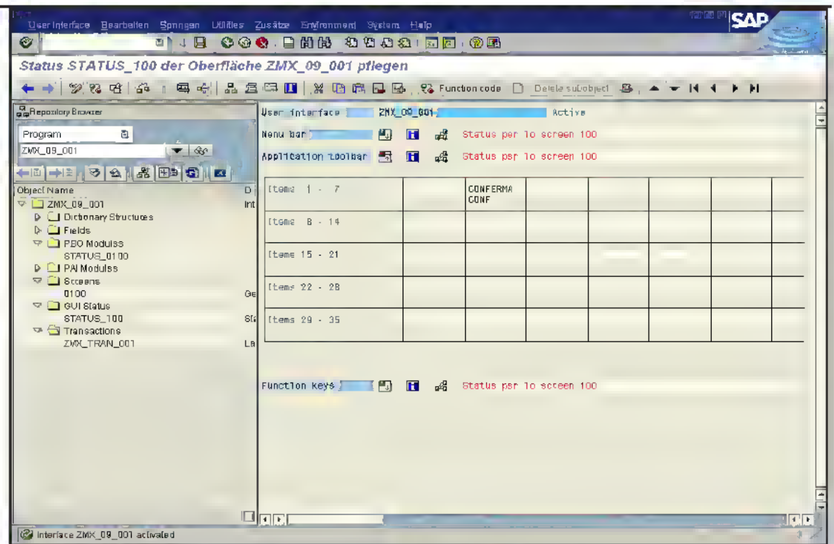


Figura 5.

Figura 6.

File Edit System Help

SAP

CONF

Codice

Nome

Cognome

Conferma

Annulla

BBS

Posta

A colloquio con i lettori

Da Fabio.

...

in questo momento in cui la crisi economica si fa sentire a tutti i livelli, mi sono chiesto se il recupero dei vecchi sistemi di calcolo o quantomeno l'uso più efficente di quelli attuali, possa in qualche modo mitigare gli effetti economici della crisi. Cosa ne pensate?

Rispone Salvatore.

A leggere i giornali e sentire le prese di posizione dei vari esperti di economia, la risposta alla crisi economica starebbe nel rilancio dei consumi. Ma anch'io mi sono fatto la stessa domanda del nostro amico lettore. Infatti sembrerebbe più logico ridurre i consumi, ad esempio tirando avanti con il vecchio PC, piuttosto che cambiarlo al solo scopo di aumentare vendite e, non dimentichiamo, i profitti di coloro che sono sulla filiera. Questo naturalemnte dal punto di vista del bilancio famigliare.

E' vero che il nostro stile di vita, basato praticamente solo sul consumo, assicura a noi occidentali una vita agiata o quantomeno dignitosa a tutti i livelli, compresi operai, commessi, impiegati, etc..., cioè a coloro che stanno alla base della piramide produttiva. E' però altrettanto vero che la sperequazione fra i ceti sociali, peraltro sempre esistita, si sta facendo sempre meno sopportabile a mano a mano che la forbice fra povertà e ricchezza si allarga per effetto della crisi economica.

Che se ne dica la crisi economica la stanno pagando i ceti inferiori, fino al cosidetto "ceto medio" che risulta il più penalizzato. Tocca invece poco i professionisti e gli imprenditori che tuttalpiù non registrano i guadagni del boom economico o al massimo chiudono le loro attività conservando per se un discreto gruzzoletto, depositato magari in "luoghi sicuri".

Da una parte quindi sembrerebbe un buon suggerimento quello di essere parchi nell'uso della tecnologia e soprattutto nel rinnovo della stessa. Questo ci garantisce un minimo di capitale di risparmio alla fine dell'anno, da conservare per tempi migliori (o peggiori, purtroppo), ma non condanniamo con questo comportamento coloro che lavorano agli strati più bassi della filiera produttiva (ormai in Italia si produce poco in questo comparto) e soprattutto commerciale?

C'è anche da considerare un atteggiamento che è duro a morire e che è quello scatenato dal deside-

rio del possesso a tutti i costi degli ultimi gadget tecnologici, per non parlare di altri status symbol come l'automobile...

Il retro computerista non è certo la persona che può incidere in maniera significativa sull'andamento del mercato tecnologico. In primo luogo siamo talmente in pochi al punto che se fossimo una specie animale saremmo considerati in estinzione, e dall'altro per sua impostazione mentale l'amante del vintage è poco attratto dalle novità e tende ad usare tecnologie vecchie, anche per le proprie necessità quotidiane che esulano dall'hobby.

Io sto scrivendo su un Macbook del 2000: proprio non ci penso nemmeno di prendermi un portatile di nuova generazione, per quanto riconosca che questi ultimi hanno prestazioni che il mio fedele compagno di lavoro si sogna. Certo sarà costretto prima o poi a cambiarlo perchè ad esempio si guasta il display (l'ho giò sostituito una volta ma per fortuna era in garanzia perché il costo superava i 600 Euro!). In ogni caso lo sto usando più di qualsiasi altro proprietario di portatile Windows mai si sarebbe sognato. Questa è la differenza, secondo me, cioè quello che io chiamo l'uso "parco" della tecnologia.

La condizione mentale che porta a questa frugalità non è purtroppo molto diffusa e soprattutto le nuove generazioni le abbiamo cresciute all'ombra del totem del consumismo più sfrenato e senza che ci sia stata una corrispondente crescita culturale.

Il benessere, i consumi e i conseguenti maggiori introiti erariali, non sono stati utilizzati da nessun governo, destra o sinistra che sia, per investimenti nel settore cultura; basta vedere come sono ridotte le scuole di ogni ordine e grado. Sì, ci sono eccezioni, ma sono sporadiche e poco incidono nella media culturale del paese che non è mai stata così infima come da dieci anni a questa parte.

Possiamo anche promuovere l'uso "prolungato" delle tecnologie, non certo il rispolvero dei Commodore64, visto che le esigenze di elaborazione di oggi non sono comparabili con quelle di trenta anni, ma nemmeno di dieci anni, orsono.

A questo proposito vedo molto bene le iniziative di trashing e la diffusione del software Open, come Linux ad esempio. Infatti non è tanto che Linux possa funzionare bene su hardware datato, cosa peraltro possibile, che mi da questa convinzione, ma il fatto che se si riesce a scrollarsi di dosso la logica del "sempre di più anche se inutile" della tecnologia Windows, automaticamente si otterrà un utilizzatore migliore. Intendo con questo dire che sarebbe meglio usare quello che abbiamo fino in fondo piuttosto che rincorrere improbabili miglioramenti che non garantiscono una vera crescita di produttività.



La prova hardware ospitata nel prossimo numero sarà una sorpresa: non un computer ma una macchina speciale, ovviamente computerizzata... vedrete!

Parleremo poi del recupero dei floppy per Apple della serie II; ci sarà un racconto della serie "in azienda" scritto dal nostro collaboratore letterario Bes; ci occuperemo anche dei numeri romani scoprendo qualcosa che forse non tutti conoscono.

.... e molti altri argomenti vi aspettano sul numero 23 di Jurassic News, l'unica rivista di retro computer italiana.